ISSN 1351-1343

QReview

£2.50

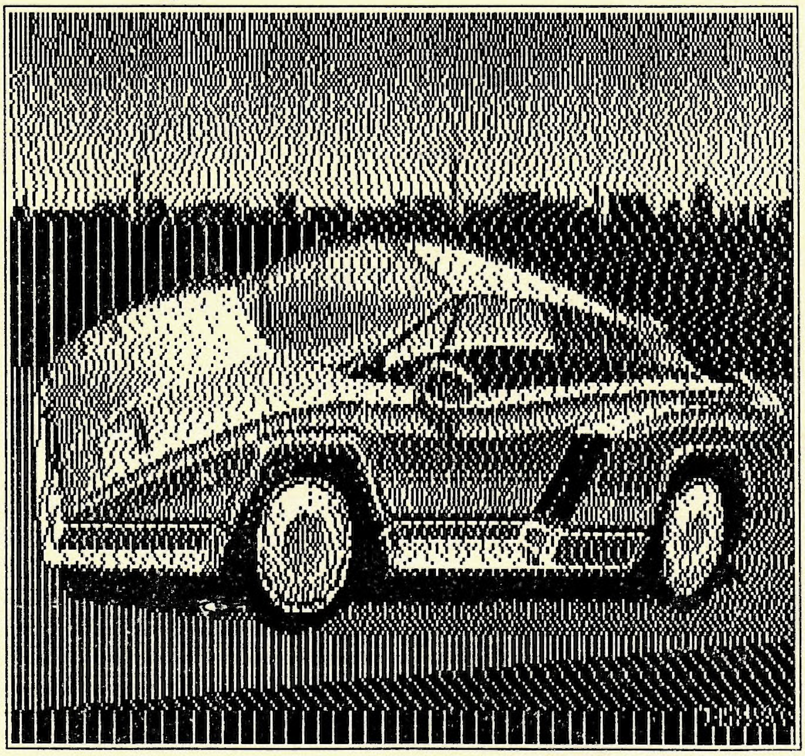
News, reviews and articles for the QL and compatibles

Volume 2 Issue 3

March 1995

OPEN WORLD

INTERNATIONAL QL NEWS



HARDDISKS & SOFTWARE

FREE COVER DISK

C68

BEYOND JUSTICE

FUNCTIONALITY

QD7

THE FORTH DIMENSION

PLUS MUCH MORE ....

QReview

(ISSN 1351-1343)
is published by
© Quo Vadis Design
57 Shaftesbury Road
Romford Essex
RM1 2QJ
Tel/Fax: 01708-755 759

Editor: Bruce Nicholls

QReview is published quarterly and is available by subscription for four issues, at a cost of £10 in the UK, £12 in Europe and £17 any other country. Back issues are available from the publisher. Please make payments payable to 'Quo Vadis Design'. Payment in UK pounds Sterling currency only, cheques (drawn on a UK branch of a Bank or Building Society), Postal Orders, International Postal Orders and Eurocheques are all accepted. Prices quoted above are valid until 31/12/95.

Editorial enquiries should be sent to the address given above. We welcome letters, comments, suggestions and articles from our readers. The submission deadline for the next issue is the 10th of June 1995.

Articles submitted for publication should be on a 3.5 DD or HD disk, or microdrive, in either Quill, Perfection, Text87 or Editor format and be accompanied by a printed copy of the article. Saved Screen dumps produced with the Sbytes command are also accepted together with screens saved by most QL drawing packages (Please indicate which package you used to produce the graphics).

Advertising rates are £25 for a full A4 page, £15 for half an A4 page and £10 for a quarter A4 page. Discounts are given for multiple page adverts.

The articles/reviews in QReview are independent and not affiliated to any vendor of hardware, software or services.

While every care has been taken with the compilation of this publication, we cannot accept responsibility for any errors or omissions. If any error or omission is found, we will endeavour to publish clarification in the next available issue.

The opinions expressed are those of the authors and are not necessarily those of the publishers/editor.

QReview is produced on a QXL using a HP Deskjet 500 printer. The software used to produce the magazine is Text87Plus4 Publisher (Software87), Linedesign (PROGS), Image Processor (QVD) and Screen Snatcher (QVD).

CONTENTS ...

ADVERTS

--- NEWS ---

DI-REN TO REPLACE QL-PC FILESERVER

Following incompatibility problems with the Super Gold Card, DI-REN have withdrawn their product QL-PC Fileserver from general sale. A replacement version, QL-PC Fileserver II, is currently under development and will be available in the near future. The new version will include additional features, improved performance and compatibility with the Super Gold Card. One of the new features allows long QL names to stored on the PC.

The new program will be available as a low cost upgrade to existing QL-PC Fileserver owners. Users wishing to receive upgrade information should write to DI-REN, at the address below, stating that upgrade information is required and include their name and address.

QL-PC Upgrade
DiRen
59 William St
Walsall
West Midlands
WS4 2AX
UK

ITALIAN INTERNATIONAL MEETING

The Italian International meeting will be held on Sunday 7th May 1995. The venue is c/o Sala Congressi Circoscrizione 2, Via F.lli Cervi, 70, Pieve Modolena - Reggio Emilia, Italy.

For more details contact Davide Santachiara, Via Emilio De Marchi 2, 42100 Reggio Emilia, Italy. Telephone (Italy 0522 300409).

SOLVIT-PLUS2 STILL AVAILABLE

Solvit-Plus 2, an electronic dictionary for the QL and compatibles, previously available from DJC will be available direct from the author Geoff Wicks.

There will be a period of about 2 months where there will be no active promotion of the program for practical and legal reasons. Small changes will be made to the code following suggestions in a review of the program in the last issue of QReview. There will be a rationalisation of dictionaries to take into account the development work which has continued after the publication of Solvit-Plus 2. The default English dictionary will have 64,000 words instead of a 37,000 word dictionary.

The author would like to hear from any QL user who can supply word lists in Spanish or in any Scandinavian language.

For more information contact Geoff Wicks, Bertrand Russellstraat 22, 1097 HL Amsterdam, Netherlands.

JOCHEN MERZ SOFTWARE UPDATE SMSQ/E

At the time of writing the latest version of SMSQ/E is V2.51. The latest changes include the possibility to change the system default character set and improved serial port reliability.

The ability to change the system default character set allows the user to replace the default fonts with any QL character set. Once the font is loaded any program executed afterwards will use the new font. This includes any program which uses the standard screen driver e.g. 99% of software. Problems with using the serial ports, loosing characters, with the Goldcard and Super Goldcard has also now been fixed.

--- NEWS ---

MIRACLE TO SELL SOFTWARE

Miracle systems will be selling Pointer Environment software from QJUMP and PROGS. The software includes QPAC2, QPAC1 and LINEdesign. For more information see page 24.

QUO VADIS DESIGN BRANCHES OUT

Quo Vadis Design has branched out and is now selling a large range of software previously available from Dilwyn Jones Computing. For more information see the back cover.

DI-REN RELEASE SYSTEM AMADEUS

Di-Ren have fully released their Amadeus System. The interfaces currently available are a QL controller, Centronics controller (PC), Centronics interface and the Ama-Sound Interface. For more detailed information write to Di-Ren , 59 William St, Walsall, West Midlands, WS4 2AX, UK.

JOCHEN MERZ SOFTWARE RELEASE PIPES

A new Game has been released by Jochen Merz Software called PIPES. The idea of the game is to connect various pipes together in a field before the flowing water escapes.
The program costs DM 29,- plus P&P {DM 9,- (Europe) or DM 14,- (Overseas)}. It is available from Jochen Merz Software (Germany 0203 501274), Im stillen Winkel 12, 47169 Duisburg, Germany.

IN THE NEXT ISSUE ...

Datadesign Datadesign API
PFdata Genealogist 3
The Forth Dimension Archive P.A.Y.E.
Interrupt Plus Lots more ....

COVER DISK

The cover disk that is included with this magazine contains three distinctly seperate items. These items are a demonstration of Plus4 the word processor from Software87, an index to QReview which runs under Archive and a catalogue and sample programs from QUBBESoft P/D. To be able to fit all of these onto one DD disk the programs have had to be supplied in a compressed form. To uncompress the programs you have to run the boot program supplied on the disk and follow the instructions given, this will only work if you have TK2 activated. If you have problems with the above please contact Bruce Nicholls on 01708 755 759.

QREVIEW INDEX

Once you have uncompressed the QReview Index full instructions on how to use it are given in the Quill file README_DOC.

PLUS4 DEMONSTRATION

Once you have uncompressed the Plus4 Demonstration you have to place the disk in drive 1 and type the following command: EXEC flp1_plus4demo. On a QL without the Pointer Environment, after the prompt appears on the screen hold down the <CTRL> key and press <C> to move the cursor to plus4. You then have to press <L> for Load and type in FLP1_demo and press enter. This will present you with a small manual on how to use the plus4 demo.

****

SPECIAL OFFER

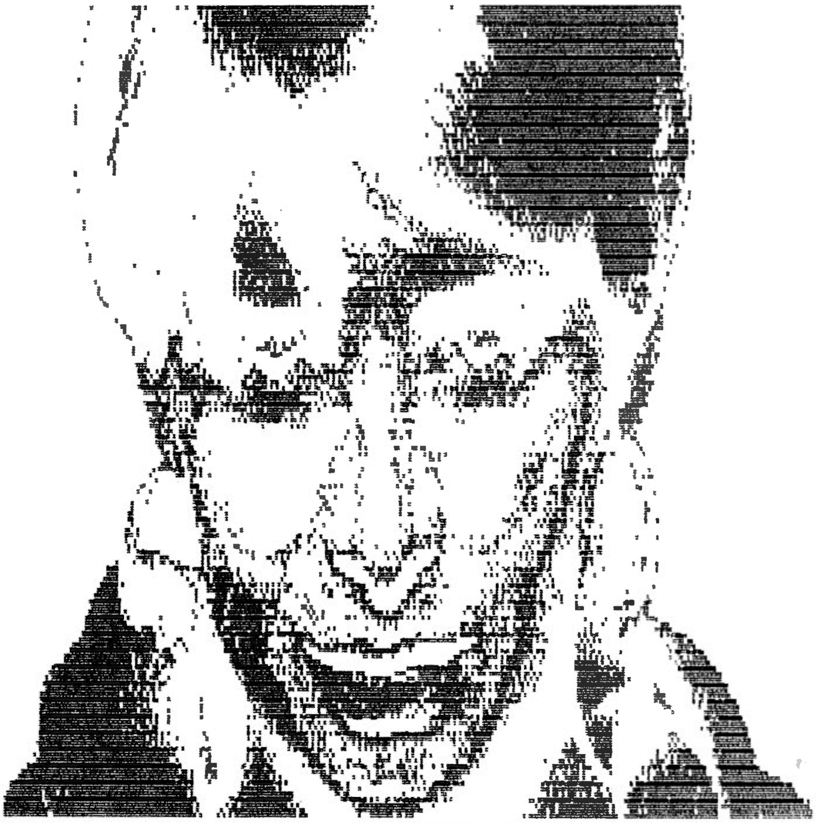
We have arranged a special offer with Software87, if you would like to purchase Plus4 you can buy it at a special price of £59 instead of the normal price of £79, you can also purchase the Plus4 Publishers pack which includes Linedesign at a special price of £179 instead of the normal price of £199. To take up this offer you must write to Software87, before May 31st 1995, quoting the QReview special offer. Software87, 33 Savernake Road, London, NW3 2JU. Tel 0171 485 9008 (2-5pm weekdays).

****

QUBBESoft P/D

Once you have uncompressed the QUBBESoft P/D programs the instructions are given in the Quill file README_DOC.

CONVERT PCX



16 colours - RECOL command

To a QL user who knows only one simple command for saving a screen - SBYTES flp1_screen,131072,32768 - the PC has a bewildering number of graphic formats. A PC program that I possess allows you to convert graphics screens between no less than 16 different bit map formats.

The enthusiastic user of QL DTP software cannot avoid coming into contact with PC graphics. Although there is now an abundance of clip art in QL format most of it has been transferred from the PC. Two or three years ago I managed to transfer some simple PC clip art using the multitasking possibilities of Conqueror, but it was a long and slow process which gave images that were squashed horizontally. A clip art picture of a bowler was turned into a sombrero!

In the last few years software has been written for converting PC images into QL format. Open World from Ergon Development transfers GIF, TIF, IFF (Amiga) and CUT files [see review on pages 9-10] , and Convert-PCX from Quo Vadis Design transfers PCX files to QL format. Conversion programs for graphic images cannot be easy to write. Some PC pictures have 256 colours, compared with the QL's 4 in high resolution mode, and the different screen resolutions make it difficult to get an image in the right proportions after transfer. Both Ergon Development and Dilwyn Jones, the author of Convert-PCX, are to be congratulated on their achievements.

The present version of Convert-PCX can only convert monochrome pictures or those with a maximum of 16 colours. This, as I shall explain later, is not a serious restriction. Like Open World Convert-PCX cannot read MS-DOS discs directly, so that any PCX picture has first to be transferred to a QL disc using a program such as DISCOVER available from Dave Walker or XOVER from Digital Precision. When you have done this Convert-PCX is quick and simple to use.

I put it to some hard tests. The first couple of times I used the program I had deliberately not read the manual. This proved to be no problem as sufficient on-screen information was given. I multitasked it with Xover without any problems, although a command for redrawing the screen would have been useful. (Hint: use ESC and take it from there.) I slipped a couple of files with 256 colours in between the others, but Convert-PCX ignored them. I pretended GIF files were PCX files, and, again, Convert-PCX was not fooled. I gave it output from a PC Graphics program with which Open World sometimes has difficulty, but Convert-PCX handled it without problems. It was also quick. I converted and viewed 5 files in less than a minute on a Gold Card QL.

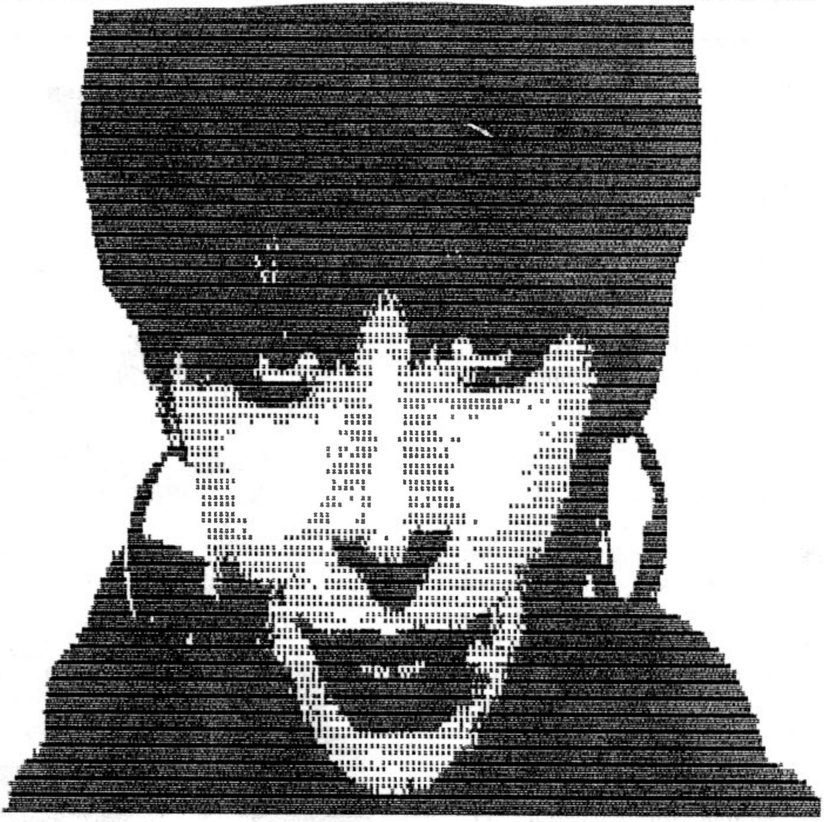
The Convert-PCX manual says that after a brief introductory screen you are "confronted with a "menu" and confronted is an appropriate word, because it requires answers to no less than 9 questions over drive numbers and file extension names. Complete this menu and you are then confronted with another menu in which you have to choose between 9 conversion options. When you have chosen one of the options, you are asked a few more questions. Put this way it sounds very off putting, but in fact the program is simple to use.

PC screens have a higher resolution than the QL screen and this means that many pictures will not fit on a QL screen. Convert-PCX has several ways of getting round this problem. The screen can be saved as a page of one of the Page Designer desk top publisher programs; it can be reduced in size; it can be saved as a number of separate screens; or a part of the screen can be saved. It is because so many options are offered that the menu screens appear fearsome. In practice you will use only a couple of the options regularly, and will discover that you do not have to answer all the questions on the menu.

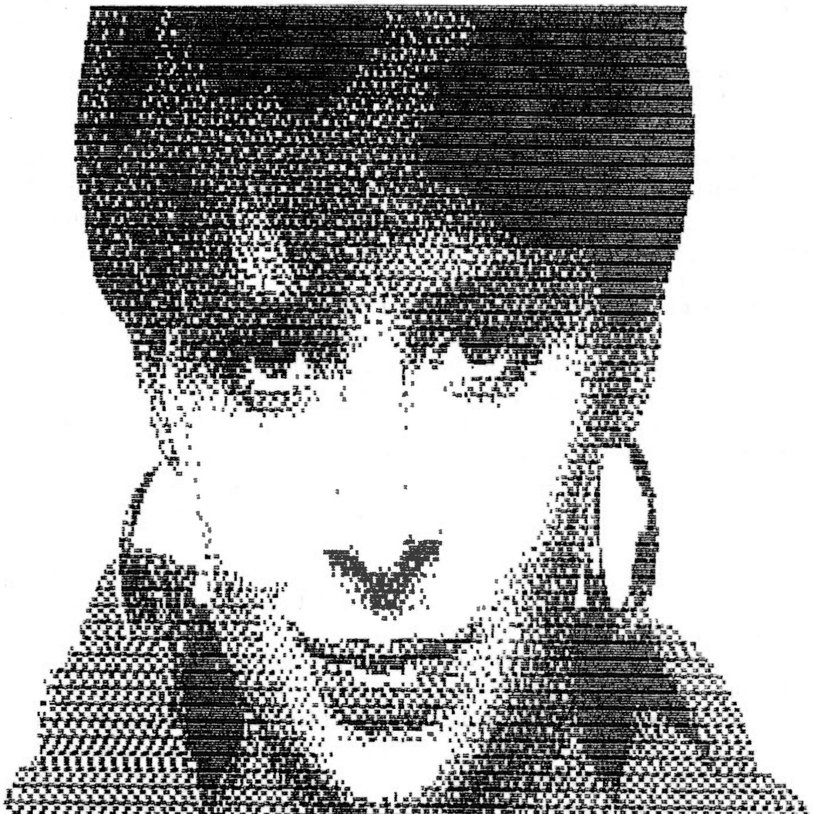
PC screens also have more colours than QL screens. When 16 colours are reduced to the 4 of the QL's high resolution mode the result is not a pretty sight. The image is usually unrecognisable. I used as test material a portrait of a woman that had appeared on the cover disk of a PC magazine. I first used a PC graphics program to reduce it from 256 colours to 16. After conversion to QL format the poor woman had a bright red and green face. Unlike Open World Convert-PCX has no routine for converting a colour image into monochrome, so I tried both the QL's RECOL command and Professional Publisher's mode 4 texture routine. The results are shown in the accompanying illustrations. It can be seen that the picture is hardly recognisable.

I then tried reducing the picture to a grey scale in the PC Graphics program before converting. However what is a grey scale for a PC is still a coloured picture for the QL so that I again had to use the RECOL and mode 4 texture conversions. The result was better than with the 16 colour picture as the image is recognisable, but hardly attractive.

Finally I dithered the picture, a process for making a monochrome image of a colour picture, in the PC Graphics program before converting

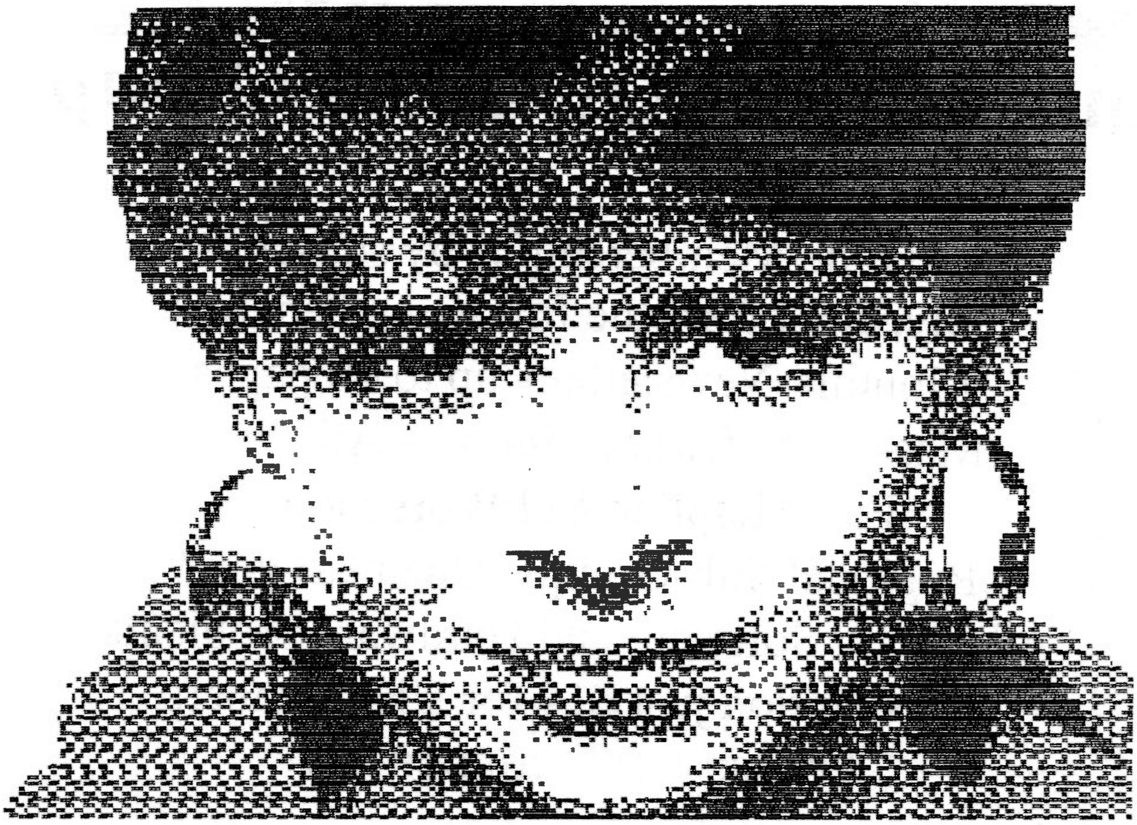


16 Grey Scale - RECOL



16 Colours - Dithered

CONVERT PCX



The Final Version

using Convert-PCX. At last success although the dithering from a 16 colour image gave a much better result than dithering from a 32 colour image. There was one further problem, the image was slightly compressed horizontally. I therefore used Professional Publisher's graphics routines to expand the image once horizontally and then compressed it twice horizontally. There was surprisingly little loss of quality and the picture was more or less in the right proportion.

In summary Convert-PCX is a program that I can highly recommend, although it is really only suitable for monochrome images. For colour pictures I would strongly advise the use of a PC graphics program to dither the image before conversion to QL format. Dilwyn Jones suggests the use of Graphics Workshop which I can also recommend.

Earlier this year Jochen Merz published a list of QL users' wishes for their computer. Number 12 on the list was a scanner. QL users with access to a PC may find it worthwhile to consider buying a cheap PC scanner. They can then use a cheap, possibly public domain, PC graphics program to change the pictures to PCX format and then Convert-PCX to transfer the pictures to the QL.

Geoff Wicks

Convert-PCX is available from Quo Vadis Design (01708) 755759, 57 Shaftesbury Road, Romford, Essex, RM1 2QJ, UK. The price is £10 plus P&P (UK free, overseas add £1).

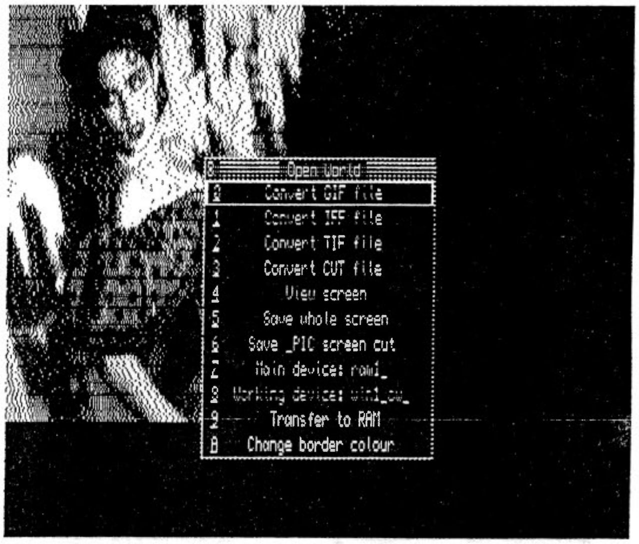
XOVER is available from Digital Precision Ltd, 222 The Avenue, London, E4 9SE, UK. It is part of the QL Collection available at a price of £179.

Discover is available from Dave Walker (01707) 652791, 22 Kimptons Mead, Potters Bar, Herts, EN6 3HZ, UK. The price is £20.

OPEN WORLD

'This package was named Open World because its aim is to allow the QL to import graphic images created on other computers'

The above extract from the introduction section of the manual for Open World neatly sums up the objectives of this program from Ergon Development. Why, I hear you cry, would you want to convert images from other computers to the QL ? The plain and simple truth is that the QL, for all its technical prowess, has been rather limited by its graphical capabilities both in hardware and software. This situation has now started to change with new powerful graphic software becoming available and the imminent release of Miracles Graphics card but the QL has a lot of catching up to do. There is an abundance of high quality images available in the Public Domain for other computers and this program allows the conversion of these images to a QL format.



The first problem of converting these images (apart from where do you get these images?) is how do you read a foreign disk format ? The answer my friend is blowing in the wi.. well the answer is to buy SMSQ/E as this is able to read MS-DOS disks directly. Ok that may be a bit of an overkill to transfer a few files but think of the benefits a fast multitasking SBASIC etc... Well if you don't fancy SMSQ/E you could always buy MultiDiscover, available from Dave Walker, or the QL Collection, available from errr DP, which contains Xover and Super Media Manager.

Once this small problem has been overcome you can then get stuck into the real job of converting files. Loading Open World is a simple matter of either running the supplied boot program or manually loading in two extension files and then EXECing the main program.

When the main program is started you are presented with the Main Menu containing eleven choices, as per one of the screen shots on this page. To choose the option you require you simply move a highlighted box over your required choice, using the cursor keys, and press enter. The manual explains all the short cut keys together with the menuing system that is used very concisely and in a pleasantly readable manner.

The first four options are for converting the four different graphical formats that the program can convert, of which more later. These are GIF, IFF, TIFF and CUT which are four very popular formats. The other options are described below:-

View Screen - this allows you to view a screen that has been converted on the screen without any menus being present.

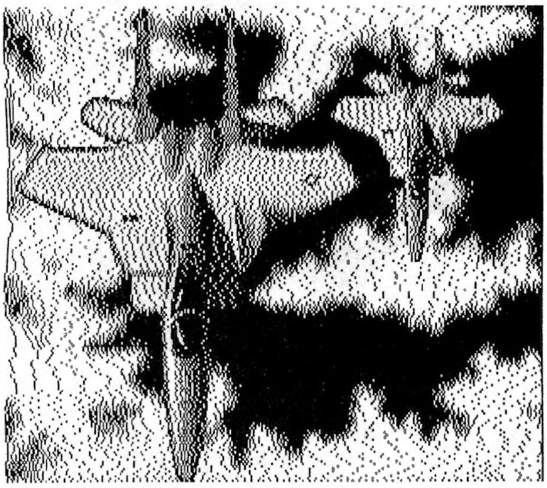
Save Whole Screen - This saves the whole screen image (including extended screen sizes) as a normal QL screen dump to a file.

Save PIC Screen Cut - This option allows you to select the size and shape of the converted screen image and saves it in Pointer Environment format (PIC) so you can later use it in a graphics program that supports PIC format.

Main Device - The main device is the device that contains the Open World program and conversion programs. Open World works by executing the main menu program, written by Davide Santachiara, then when a conversion needs to be done executes a C conversion program, written by Giuseppe Zanetti. This is why the program needs to know what device the programs are on.This can be FLP1_ or a sub-directory on a hard disk.

Working Device - This is the device, or subdirectory, that contains the foreign format files. The files have to end in _GIF, _IFF, _TIFF and _CUT for the program to recognise them.

Transfer to RAM - As I said before the main program works by loading the conversion programs separately when needed. This has the undesirable effect, if using floppy drives, of being unable to swap disks when using the program. This option copies all the necessary files to a ram disk therefore freeing up your floppy drive and saving wear and tear providing you can create a ramdisk.



Change Border Colour - This option surprisingly allows you to change the border colour of the main window.

Once you have set up the program so it knows where to read the images to be converted you can start the conversion process. This is acheived by selecting the type of conversion from one of the four formats the program knows about. This presents you with a list of files that have the same extension ie. _GIF, you then press enter on the file you want to convert. This then brings up an information window detailing the information contained in the graphic file such as size, colours etc.

The next menu presented on pressing enter is the Image Loading menu. This contains the following items:-

Process File - This starts the conversion process.

Output - This displays your screen size.

Dataspace - The conversion program for the GIF image needs to load the whole image into memory, this option specifies how much memory is allocated for this option.

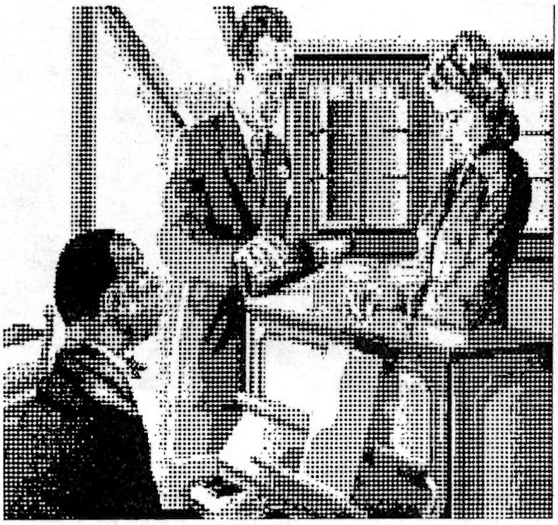
Loading in Mode 4/8 - This selects the mode for the conversion.

Dithering On/Off - Dithering is a method of changing colours into grey levels so the output can be viewed on a monochrome screen. This also produces a very good black and white image for printing.

X/Y loading offset - If you have an image to convert that is larger than your screen size you can select the starting positions for the conversion process to start from. This enables you to convert the picture into two screens and them later join them using a QL graphic program.

Set Default values - This resets the menus values to there original defaults.

How does the program perform ? Most admirably is my answer, the program converts files with ease. The samples shown on this page show how well the Dithering works on colour images. The speed is not great but the quality of the image is well worth it. Being able to use extended screens is also a joy as you can convert and see a large image in one go and it does work on the QXL. The only thing missing from the program is the ability to convert large groups of images automatically. There is also an added bonus of a PC disk which contains utilities to allow the viewing and conversion of QL screens on PCs, Sun and Vax machines together with the source code. For anyone using the QL for DTP this program is a must.



Derek Fish

Open World is available from Ergon c/o Davide Santachiara (Italy 0522 70409), Via Emilio De Marchi 2, 42100 Reggio Emilia, Italy. The price is £18 fully inclusive (make cheques payable to D.Santachiara).

C68

Introduction

The C compilation system for QDOS/SMS has recently been updated with a new version available. Release 4.14c is now released and on general circulation. It should be available via all normal C68 distribution channels.

The 4.14 release of the C68 system was first made available at the Quanta workshop on 20th November, 1994. In normal predictable fashion just after the release had been done the first bug reports came in from those who had early beta copies! A patch was therefore produced to fix these problems. If you have the original release (for which the version number of the LIBC_A library was 4.14) then you are advised to get the patch to upgrade it. With the patch applied, the version number of LIBC_A will be 4.14c.

What has Changed

Fundamentally this release is as much a "bug fix" release as anything else. However there are quite a number of changes that will be of relevance to the serious programmers.

The key features of this release are:

-All known bugs in the libraries and compiler have been fixed.

One particular area that has received special attention is those routines that handle characters that are outside the normal ASCII character set. A number of changes have been made to handle such cases correctly.

-The main C68 compiler phase has had significant internal changes made to it. The result is that code is both smaller (typically around 3-5 %) and faster.

There are also a number of new warning conditions that are detected and reported on. In addition new error conditions are now detected that previously were being missed.

-The LIBC_A library has a large number of new routines added to it. The majority of these are related to supporting functions commonly found in Unix environments (and thus helping those porting programs from such environments) although several of them are of a more general application.

-There is a new "minimal" start up module mode. This allows those who are using C as a "high level assembler" and who do not need the full C environment because they are primarily calling underlying operating system capability to significantly reduce the size of their programs.

-Name hiding within the libraries has been more extensively implemented. Again this should be of particular help to those porting programs from other operating systems although it will be invisible to the average user.

- Various very commonly used library routines have new significantly faster versions.

The net result is that this release is a worthwhile upgrade for the serious C programmer whatever previous release they were using.

More casual users may decide that they can skip this release - particularly if they are using the 4.13 release. Such users may be better of waiting for the 4.20 release planned for the New Year (see section on Futures) before they bother upgrading.

Utilities and Programming Tools

Possibly of as much significance as the 4.14 release of C68 is the fact that at roughly the same time, a number of programs that fall into the category of utilities or programming tools have also been made available for QDOS/SMS for the first time.

GNU Revision Control System (RCS)

The RCS system is a must for any serious programmer - whatever programming language they are using. It consists of a suite of programmes that provide tools for keeping track of all the changes that you make to your source files.

The facilities that are provided with RCS are too extensive to summarise here - so all I can suggest is get a copy and read the detailed documentation that is provided with it. Please note that to use RCS you will also need the GNU Diff package mentioned below.

GNU Diff Utilities

This is a set of programs that are used to help in finding the differences between two files - with particular emphasis on use with text based files.

These programs differ from simple compare programs in that they have fancy algorithm's for re-synchronising between files that are being compared after a difference is found.

THE FUTURE

You might be interested in knowing in what direction future developments of the C68 system will go. On the basis that these are only intentions and not promises, here are the predictions for future developments.

The 4.20 Release of C68

The plans that are firmest are those for the 4.20 release. This is likely to appear sometime around the end of March 1995. As well as the normal incremental enhancements to the libraries and fixing any bugs there are two major new facilities planned for this release.

Run Time Link Libraries (RLLs)

The development of the Runtime Link Library system was started some time ago by Lester Wareham, and is now being completed by Dave Walker. The development is very near completion - so much so that originally it was hoped that it would make the 4.14 release. Sadly it was not ready in time, so it will be one of the main features of the 4.20 C68 release.

If you want to know more about what the RLL system is and the facilities it will provide, then the preliminary documentation for the RLL system has been included with the C68 4.14 documentation disks.

Signal Support

Richard Zidlicky from Germany has been working for some time on developing a package for emulating the Unix signal system. He has just recently made available a preliminary release. A significantly enhanced version of this facility is being developed and will be integrated into C68. Again this facility should be part of the 4.20 release.

Support for Unix style signals will hopefully make it possible to port Unix style programs that were previously difficult to port because of their reliance on support for signals. A typical example would be a Unix style shell to QDOS/SMS as an alternative to the current SuperBasic based command interface.

Future Release of C68

As we move further into the future, then plans will be more subject to change. However, there are a number of areas in which work is already far enough advanced to say with some confidence that the capability will be supported by a future release of C68.

ENV system v2

This is a new version of the Environment variable support that can be used from any language, and not just C and SuperBasic as the current version. The revised implementation will also put a smaller overhead on C programs thus reducing the size of the average C program.

Hardware Floating Point Support

The current C68 implementation only supports software floating point. The advent of systems such as the QXL means that there are now QDOS/SMS systems that have floating point hardware. The intention is that the C68 system will detect such hardware if present and use it in preference to the software version with attendant increases in speed.

New tools and Utilities

There is a whole host of software tools that are either currently being ported or being considered for port to QDOS/SMS. Examples are:

- GNU Textinfo documentation system
- NROFF
- LINT

C++ Support

We are regularly asked whether C68 will be enhanced to be a C++ compiler. It is something that has been looked at and some preliminary work has been done to try and estimate the effort that would be involved. I am afraid that the conclusion that we have come to is that it is very unlikely that C68 will ever be enhanced to a C++ compiler as it looks like being a mammoth undertaking.

If C++ support ever does appear on QDOS or SMS it is much more likely to be as a result of someone porting the GNU C++ public domain compiler. If this port is ever done, then the role of C68 is likely to be limited as a tool for helping to do this port.

SUMMARY

The C68 system for QDOS and SMS is still being actively developed. However, new ideas are always welcome so please feel free to make suggestions (or criticisms if you must!). It is only by active participation and feedback from users that we gauge whether we are developing C68 in the right direction, and what users think of what we have done so far.

Dave Walker

Dave Walker (01707) 652791, 22 Kimptons Mead, Potters Bar, Herts, EN6 3HZ, UK.

QUBBESoft P/D

38, Brunwin Road, Rayne, Braintree, Essex. CM7 5BU.
Tel: (0376) 347852 Fax: (0376) 331267
24hr - 7 Day Week Service.

PD & Shareware Software

QL Emulator AMIGA V3.21 (3 Disks)	£3.00p
COWO QL Screen Collection (3 Disks)	£3.00p
C68 Compiler Runtimes (2 Disks)	£2.00p
C68 Compiler Documentation (2 Disks)	£2.00p
C68 Compiler Source Code (2 Disks)	£2.00p
MicroEMACS V3.11 (2 Disks)	£2.00p
GNU Text Utilities (2 Disks)	£2.00p
Adventure '93 (2 Disks)	£2.00p
XLisp V2.1d (2 Disks)	£2.00p
STAR TREK Reference (2 Disks)	£2.00p
SPECIAL 1 Q.A.T.S.	£1.00p
SPECIAL 2 Molecular Graphics V5.11	£1.00p
SPECIAL 3 XLisp V1.4a	£1.00p
SPECIAL 4 QPACER	£1.00p
SPECIAL 5 'C' Tutorial	£1.00p
SPECIAL 6 C68 QPTR Comp V3.02	£1.00p
SPECIAL 7 CPORT Support Library	£1.00p
SPECIAL 8 QRactal Screens Demo	£1.00p
SPECIAL 9 QL Home Finance	£1.00p
SPECIAL 10 ZM1+ Spectrum Emulator	£1.00p
SPECIAL 11 Lib CURSES V1.20	£1.00p
SPECIAL 12 ELVIS Editor V1.6	£1.00p
SPECIAL 13 Text 'N' Graphics Demo	£1.00p
SPECIAL 14 'C' Programming Tools 1	£1.00p
SPECIAL 15 'C' Debugging Tools	£1.00p
SPECIAL 16 Ergon Dev Demo Disk	£1.00p
SPECIAL 17 Ger to Engl Word List	£1.00p
SPECIAL 18 QDesign Demo	£1.00p
SPECIAL 19 D.J.Walker Demo Disk	£1.00p
SPECIAL 20 DBAS Database Package	£1.00p
SPECIAL 21 FTC II Demo	£1.00p
SPECIAL 22 QL Hard Disk	£1.00p
SPECIAL 23 PSION Xchange	£1.00p
SPECIAL 24 SPECULATOR '93	£1.00p
SPECIAL 25 QLIPER Disk	£1.00p
SPECIAL 26 Line Design 2 Demo	£1.00p
SPECIAL 27 Compu-Clips 1	£1.00p
SPECIAL 28 Bison V1.21	£1.00p
SPECIAL 29 Flex V2.3	£1.00p
SPECIAL 30 QDOS Escher Demo	£1.00p
SPECIAL 31 Mini Clips	£1.00p
SPECIAL 32 QTPI	£1.00p
SPECIAL 33 QEM	£1.00p
SPECIAL 34 Carlo Delhez Software Collection	£1.00p
SPECIAL 35 GNU Diff	£1.00p
SPECIAL 36 GNU RCS	£1.00p
SPECIAL 37 to 43 R&S Clip-Art 1 to 7	£1.00p each
P.D. Disks Numbers 1 to 14	£1.00p each
QITALY Disk Magazine Issue 14 to 27	£1.00p each

New QL Hardware

QUBIDE: AT/IDE Interface, allows connection of modern AT/IDE Hard Drives to the QL. Plugs into the QL's expansion slot, existing Expansion system plugs into QUBIDE's through port. Works with Super Gold Card, Gold Card, Trump Card and most Memory Expansion Systems. Minerva and Hermes compatable. £ 65.00p

QL Mini Tower System: Have your QL, Disk Drives, Hard Drive and Memory Expansion all put into a PC Mini Tower Case. No more problems with Power as everything runs from PC Power Supply Unit. The drawback is you lose the use of your Microdrives. Give your QL that professional look. Enquiry Form available. S.A.E.

Fast-Net: Local Area Network, enabling connection between QL & QL, QL & ST, ST & ST (ST must be running QDOS). Fully compatable with existing hardware expansion systems including SGC & GC. £100.00p

Each Additional Fast-Net Unit: £ 55.00p

Gold Card: (Re-Conditioned) 68000 @ 16mhz, 1920K RAM, TK2, Disk Interface, RAM Disk, Printer Buffer £100.00p

Trump Card: 768K RAM, TK 2, Disk Interface, Dynamic RAM Disk, Printer Buffer. Includes A4 Manual £ 45.00p

Twin 720K Disk Drives: Fully Cased with internal Power Supply Unit £100.00p

Twin 1.44mb Disk Drives: Fully Cased with internal Power Supply Unit £150.00p

1 Year Warranty on all New Hardware
(P&P: UK + 5%- Max £5.00p, Euro + 10%- Max £10.00p, R.O.W. + 20%- Max £20.00p)

New Products in Pipeline

Back-Plane: 3 slotted Back-Plane for use in PC Mini Tower Case. Included on Back-Plane will be PC style connector for power.

DIY Mini Tower Kit: This Kit will include all the parts required to place your QL into a PC Mini Tower Case. Included in the Kit will be a Back-Plane and full fitting instructions.

Payment Methods

Cheques (in UK pounds drawn on UK Bank), Postal Orders, International Postal Orders & EuroCheques (in UK pounds) payable to:- **QUBBESoft P/D**

Cash in GB Pounds is also accceptable.

P&P Guide for PD & Shareware

Qty	UK	Euro	Z1	Z2	Qty	UK	Euro	Z1	Z2
	£ p	£ p	£ p	£ p		£ p	£ p	£ p	£ p
1 - 3	0.50	1.00	1.50	1.75	13-15	1.50	3.00	3.50	3.75
4 - 6	0.75	1.50	2.00	2.25	16-18	1.75	3.50	4.00	4.25
7 - 9	1.00	2.00	2.50	2.75	19-21	2.00	4.00	4.50	4.75
10-12	1.25	2.50	3.00	3.25	22-24	2.25	4.50	5.00	5.25

Beyond Quill

If you own Perfection and you have never done it before, try the following. Load Perfection and type some text. Now press Ctr/C to return to SuperBasic and then enter the direct command "Exec flp1_Perfection". Make this second copy of Perfection active by pressing Ctr/C and type some text. By pressing Ctr/C followed by F4, you can toggle between two copies of Perfection and SuperBasic. Multi-tasking is as simple as that!

When you are busy with your wordprocessor you will often want to refer to another program. You may wish to look up an address in a database, copy the results of calculations in a spreadsheet or even refer to another document. A multitasking wordprocessor lets you do this, Quill does not. Quill is a greedy little beast that grabs all the memory in your computer that it can lay its hands on and refuses to give it up until you leave the program. Perfection behaves more gentlemanly, only uses memory as it needs it and gives it back when it no longer needs it. It is designed to run alongside other programs. This is only possible with Quill if it is kept in control by a program such as Taskmaster, QPAC2 or Taskforce. It is a good reason for moving on from Quill to another wordprocessor.

Having said this, however, a word of warning. The other Psion programs, Archive, Abacus and Easel, are just as naughty as Quill and will not run alongside Perfection without the use of one of the three programs mentioned above. If you cannot afford to buy a commercial program, use Taskforce, the code of which was printed in QL World some time ago. All you have to do is substitute all references to "Quill" in the program with "Perfection". You will, however, have to get used to the different key presses for accessing the programs and refreshing the screens.

If you do move on from Quill to another wordprocessor, you will probably find that there is a lot of reference to the use of "blocks" in your new program. There is nothing new or complicated about blocks since Quill also uses them, although I do not think the word "block" appears in the Quill manual. If you use the COPY or ERASE commands in Quill then you are already using blocks. When you use these two commands you first have to define the part of the document which you wish to copy or erase. You go to the start of this section and press enter. You now move the cursor to the end of the required section, and you will notice, that as you do this, the background colour has changed. At the end of the section you again press enter. What you have done is to define a block and Quill has marked that block so that it is to be seen on the screen. Finally you follow Quill's instructions to carry out the required action of either deleting, moving or copying that defined block of text.

Blocks in your new wordprocessor work in much the same way. First you define the block and then you carry out an operation on it. The chances are that your new wordprocessor can do much more with blocks than simply deleting, moving and copying. Blocks in Perfection can be copied, deleted, moved, saved, loaded into the macro buffer or spellchecked. Because of the extra cursor controls and the additional speed of Perfection, block operations are usually faster and more efficient than in Quill. In some block operations in Perfection the defined block is not marked on the screen. If you find this disconcerting there is a alternative way of defining blocks so that they are visible on the screen. It is worthwhile to spend some time practising blocks in your new wordprocessor.

Searching and replacing text should also be better on your new wordprocessor. In Quill the search and replace commands are very simple and easy to use. You can only search forward from the cursor position and all searching is case independent. Search for "QL" and Quill will find "QL", "ql", "Ql" and "qL". If you wish to replace "QL" by "PC" (perish the thought!), Quill will stop at each occurrence of QL and give you the choice of replacing it or not. It will then search for the next occurrence.

A good wordprocessor will offer rather better search and replace facilities. Perfection, for example, allows you to search forwards and backwards, something that can be handy if you are at the end of your document. The searching can be either case dependant or case independent, and at each occurrence you are given the choice of replacing, not replacing or replacing all further occurrences. Even this choice can be switched on or off. Searching and replacing is an area where the speed of Perfection is impressive. Digital Precision claim to have changed all references to "God" in the book of Genesis (42,000 words) in under 9 seconds. My own benchmark is probably less impressive (and here I mean the benchmark, not the result). The longest Perfection document I have yet produced is a wordlist for my Dutch language Spellchecker dictionary. It has 39,142 words, 408,391 characters and is 634 pages long. Searching in this document is practically instantaneous.

There is, of course, a price to pay for these extra possibilities. There are more commands to learn and your wordprocessor has become a little less user-friendly. I can only repeat what I have said before. The use of these new commands is not compulsory. The wordprocessor works just as well without them. Try all the new facilities one by one. You will soon find yourself using them as a matter of course.

I had not realised how controversial spellchecking was, until Digital Precision released Spellchecker. It is an area where people have strong personal preferences. There are those who argue that you should not need a spellchecker, but should learn to spell instead. Others will say that this is an elitist argument. Some spellchecker users want to check a document when typing and want alternatives for wrongly spelt words, but others want to check a document after completion as quickly as possible. Some users want as large a dictionary as possible, whereas others argue you should use as small a dictionary as possible. Pity the poor software publisher who has to cater for all these needs!

There are many reasons for using a spellchecker. The obvious one is that there are people who do have problems with spelling. Equally there are people who can spell accurately, but who make frequent typing errors. Finally there are people, like myself, who make few spelling mistakes and can type reasonably accurately, but who are poor proof readers. If I make a mistake, I often miss it when checking through a document. I also have another problem. My native language is English, but I live permanently in a foreign country, so that I have to be bilingual. It is quite common for a Dutch word to slip into an English text or an English word into a Dutch text. Being a poor proof reader, I rarely notice it when checking what I have written.

Spellcheckers are not particularly intelligent. All they do is compare each word in a text with the words in their dictionary. If the word is not found it can mean that it is wrongly spelt or it can mean that it is not in the dictionary. Equally if a word is found it does not mean that it is correctly spelt. I once sent the editor of QL World an article in which I used the Dutch spelling "rooster" for the English word "roster". The spellchecker did not find this mistake because "rooster" is also an English word. (Judging from the letter I received back from the editor, I think she positively crowed with delight at the thought of a "duty rooster".) There is a dilemma for the software publisher. If his dictionary is too small, it will find "mistakes" that are not really mistakes. However the larger it is the greater the chance that some mistakes will not be reported, because the incorrect spelling is also a word. If I typed "QL World" as "QL Wold", "Wold" is also an English word, less common that "World". I would rather not have it in a spellchecker dictionary, unless, of course, I were a geography teacher.

In my opinion it is for the user to decide on the size of the dictionary he wants when spellchecking. Equally it is for the user to decide if he wants to check spelling while typing or only after completion of the document. If he chooses the first the speed of typing will be slower which may interrupt the train of thought, but mistakes can be corrected right away and with some spellcheckers the correction entered into the document almost automatically.

If you are one of those users who would like the spellchecker to come up with alternatives for a misspelt word, do not expect the QL to be as sophisticated as some PC spellcheckers. WordPerfect, suggests alternatives which differ from the original by one letter, or by two letters out of sequence. It then suggests words that sound similar to the original. The user pays a price for this sophistication. Spellchecking is slow and involves wear and tear on the drives unless there is cache memory available. In practice this places a limit on the size of the dictionary. The two "giants" in QL wordprocessing, Freddy Vaccha of Digital Precision and Fred Toussi of Software87, have indicated that the hardware available to the average QL user does not permit the same sophistication in QL spellcheckers. At best a QL spellchecker will look up the nearest match, display this, and let the user scroll through the dictionary until he comes across the correct word. One of my spelling mistakes is "suprise" instead of "surprise". One spellchecker suggested "supremity" and I had to scroll through 34 words until I came to "surprise". It is obvious that in some cases the correct spelling of a word would be difficult to find. Nevertheless there are many QL users who are satisfied with this level of sophistication and who are sad that it was not included as a option in Digital Precision's Spellchecker.

In this article I have tried to take a neutral view of spellchecking, but let me state my own preferences. I prefer to check a document after completion rather than during typing. I do not like having the QL constantly beeping at me. If rude noises are going to be made, I make them to the QL, rather than the other way round. I prefer to use a small dictionary and have the smallest of the supplied dictionaries installed in my system. If I come across a word marked as a mistake and I am not sure whether or not it is a mistake I can always load a larger dictionary and check again. I like Spellchecker for its speed and the clear marking of possible mistakes. (There are advantages in using the spellchecker dedicated to your system)

The choice is up to you. A good spellchecker allows the user to use it in the way that suits him and not the software producer best, and should offer:

1: A number of dictionaries of differing sizes.
2: The possibility of adding new words to the dictionary.
3: Both "check as you type" and "check on completion".
4: Alternatives to a misspelt word.

At the moment none of the QL Spellcheckers fit all these requirements. My advice is to decide what you want in a Spellchecker and then enquire if it does what you want.

When Digital Precision warn "This isn't tough, but don't try this until you have understood everything else", you can be fairly certain that many users will still find it tough. I am talking about Perfection and macros.

Macros are important timesavers in wordprocessing programs. With one key press you can insert a block of text or carry out a series of commands. Every time we use our wordprocessor we are working in what Perfection SE calls an environment. This covers among other things margins, tab settings and type of justification. The environment that I use for a letter is different from the environment that I use when writing an article. Changing from one environment to another can involve many commands. If I want to print a document, I press Ctr/P and then have to answer a series of questions. If I want to edit a Spellchecker wordlist, a SuperBasic program or a data base I need to set the left and indent margins to 1 and turn line wrap and automatic reformatting off. All these are situations where a macro could be used to save a lot of keypresses.

In the PC world, and especially with WordPerfect, macros are important. Large section of books are devoted to them; there are examples given in magazine articles; and often macros are included on the cover discs. In contrast I have seen little about macros in QL publications. You would expect QL publications to be full of hints about macros, but they are not. Do QL users never use macros? Have we taken Digital Precision's advice to heart and are still ploughing through other sections of the Perfection manual? Are we using macros, but disguised in the form of "Altkey" definitions? Are we frightened of macros? First the good news. Making a macro is not difficult. Now the bad news. Writing a macro is going to be one of the most difficult things you are going to do with your wordprocessor. I am not contradicting myself.

Let's make a simple macro. If you have Perfection load it. Press Alt/Ctr/Z to clear out any document, control codes etc. Set the left and indent margins to 1, turn off automatic reformatting and line wrap. We are now ready to begin. Just type in your address as if on a letter heading. Save this as address_macro via the export command (F3 I) to disc.

Just to show that there is no cheating, reset the QL and reload Perfection. Press Ctr/Shift/F3 to load the macro. Now press Ctr/F3. Who said that making a macro was difficult?

Let's make it a little more difficult. Maybe you would like your letter heading in bold type. How do you switch bold type on or off in Perfection? One way of switching bold on is by pressing key F3 followed by E. Normally if we press F3 it has the function of a control code that gives Perfection an instruction. It does not appear on the screen or in the document we are writing. In a macro, however, we need it to appear and the screen because we are recording key presses. Perfection has a way of doing this. First prepare Perfection for a macro as detailed above. Now press Ctr/£ followed by F3. This prints a symbol for F3 on the screen. Press E. Type your address in. Press Ctr/£. Press F3. Press O to switch off bold and save.

Of course this macro would be most useful if it placed itself at the top right hand side of the letter you are writing. Try and work out how to do this, by writing down the commands that are necessary. If you get stuck, you will find the answer in the Perfection SE manual. I hope that it is now clear what I meant when I said that making a macro is easy, but writing it is difficult. To write a macro you need to have a good knowledge of all the commands in your wordprocessor. I wrote one macro that is only 11 bytes long and I think it must have taken about 5 or 10 minutes per byte to write. What it does is remove the highlighting from misspelt words after I have spellchecked a document. If you had asked me how to do this, before I had written the macro, I would probably have replied that it was a simple process of using the "replace all" command to find all highlighted words and remove the code for highlighting. In practice this reply would be too simple. I do a lot more than using "replace all" when deleting highlighting. I do all the following things:

1: The cursor is somewhere in the document. Move it to the top.
2: Switch off Query or the macro will not work properly.
3: Press M1 A for replace all.
4: Maybe there is already something in the find buffer. Delete this.
5: Enter the control code for highlighting.
6: Maybe there is already something in the replace buffer. Delete this.
7: Press enter.

In other words, to write a macro you have to think logically and carefully what you must do to cover all circumstances in which you will be using that macro. Macros are not just difficult to write in Perfection. They are difficult in every wordprocessor. Just as there are users who have difficulty in writing a printer driver, so there are people who will have difficulty in writing a macro.

Let me end this series of articles by making a plea. We are all QL users, and part of the QL tradition is of helping one another, either through the columns of QL publications or via workshops. Perhaps it is even possible for someone to produce a (PD?) disk of Perfection and Text87 utilities.

Geoff Wicks

QD7

QD 7 is the latest incarnation of the QD text editor from Jochen Merz's prolific software house. As with most (or should that be all ?) of Jochen's products, this one requires the Pointer Environment to work. In addition, it also requires Jochen's QMENU extensions. All of the above are supplied.

QD is a text editor, not a word processor - it has no word wrap facilities or justification etc, at least, none that I could find that is. I am sure to be corrected if this information is wrong.

I have never used any versions of QD in the past and I always considered the possibility of actually buying a copy as most of my work is done under the Pointer Environment (or PE from now on, it takes too long to type it in full !) and an editor that runs under the PE would be handy - then QD 7 arrived for review - perfect timing I thought, cheers Bruce !

Before writing this review, I put QD 7 to work for some amendments that I am writing for Dilwyn's DJToolkit (oops, here we go again!) and I found it quite simple to use - which is just as well as I didn't get the full manual with the disc. It was a case of, here it is, get on with it.

Anyway, what could go wrong with a simple editor ? The answer seems to be, not a lot.

QD is loaded using EXEC, EX or can be LRESPR'd and then EXEP'd for PE experts. A command line can be passed to it specifying the file to be opened and a few other parameters so that you can, for example, load a file and then position the cursor to the first occurrence of some text within that file.

The first thing to do is load a file. This can be done using 'CTRL' and 'L' or by pulling down the FILES menu as shown in a screen shot, not a million miles from here (I hope).

Once the file has been loaded, there are more commands available, as can be seen in the screen dump (Figure 1), to allow you to insert additional files, save, save & rename the current file etc.

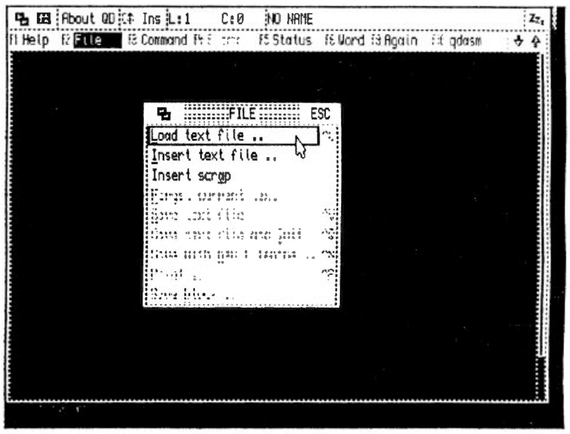
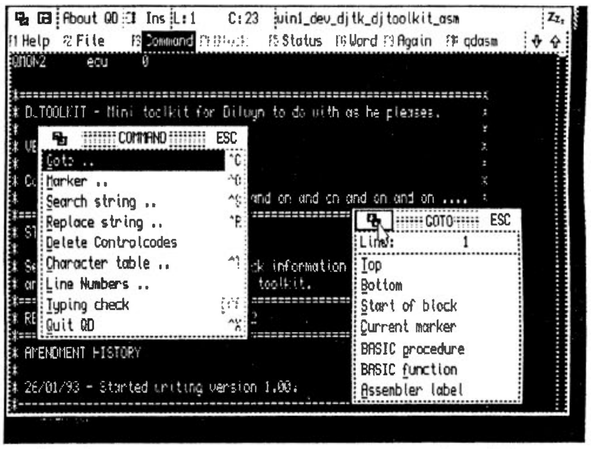


FIGURE 1



```
QMON2       equ     0
*==========================================================================
* D_TOOLKIT - Mini toolkit for Dilwyn to do with as he pleases.
* AMENDMENT HISTORY
*
* 26/01/93 - Started writing version 1.00.
```

Using QD 7 is very easy, the help that is provided is a simple text file that is loaded as required when you press F1, the QD program is of course CONFIGurable so that this help file can be loaded from anywhere that you desire. Another quick way to get help, is simply to click with the right hand mouse button - this is normally considered to be the same as pressing ENTER, but in QD 7, the two are treated slightly differently.

It seems, although I did not check everything, that all the various menu commands can be accessed via a 'CTRL something' key combination. This is a good idea for experienced users, while us learners plod on with the various menu levels. Of course there is a slight snag, but we can't blame Jochen for it, there are simply too many keys to remember !

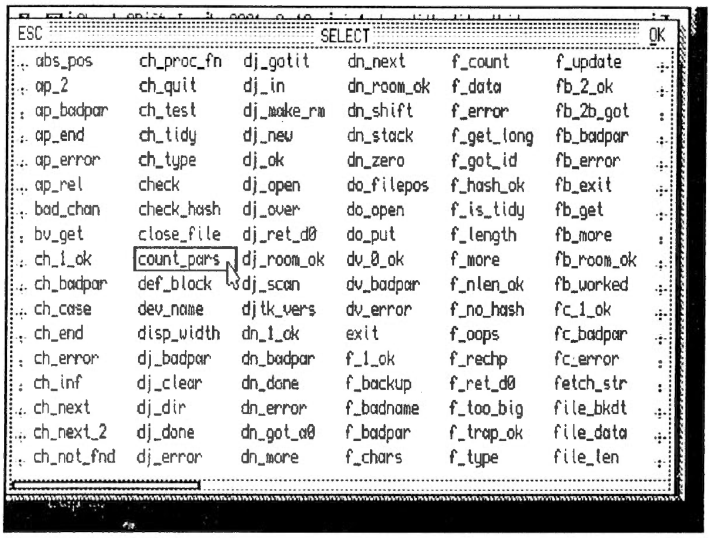
Some of the keypresses are not quite what you would expect, CTRL + L, for example will let you LOAD a file, CTRL + S, on the other hand brings up the SEARCH menu and does not do a SAVE. SAVE is on CTRL + V. As I said, it is difficult to get a suitable keypress for every command.

SEARCH and REPLACE are quite unique in QD, at least as far

QD7

as any other text editor (or word processor) that I have used is concerned. Each can have up to 4 strings that you can search for. Most other editors allow the previous search string to be the default when you next call up SEARCH or REPLACE. This is true of QL editors and PC ones too.

The SEARCH option requires that you either type or select one of the 4 strings to be looked for, then choose whether the case of the text is to be ignored or not and whether the search is to be upward in the file, or downwards. The final searching is very quick.



```
abs_pos     ch_proc_fn  dj_gotit    dn_next     f_count     f_update
ap_2        ch_quit     dj_in       dn_room_ok  f_data      fb_2_ok
ap_badpar   ch_test     dj_make_rm  dn_shift    f_error     fb_2b_got
ap_end      ch_tidy     dj_new      dn_stack    f_get_long  fb_badpar
ap_error    ch_type     dj_ok       dn_zero     f_got_id    fb_error
ap_rel      check       dj_open     do_filepos  f_hash_ok   fb_exit
bad_chan    check_hash  dj_over     do_open     f_is_tidy   fb_get
bv_get      close_file  dj_ret_d0   do_put      f_length    fb_more
ch_1_ok     count_pars  dj_room_ok  dv_0_ok     f_more      fb_room_ok
ch_badpar   def_block   dj_scan     dv_badpar   f_nlen_ok   fb_worked
ch_case     dev_name    djtk_vers   dv_error    f_no_hash   fc_1_ok
ch_end      disp_width  dn_1_ok     exit        f_oops      fc_badpar
ch_error    dj_badpar   dn_badpar   f_1_ok      f_rechp     fc_error
ch_inf      dj_clear    dn_done     f_backup    f_ret_d0    fetch_str
ch_next     dj_dir      dn_error    f_badname   f_too_big   file_bkdt
ch_next_2   dj_done     dn_got_a0   f_badpar    f_trap_ok   file_data
ch_not_fnd  dj_error    dn_more     f_chars     f_type      file_len
```

FIGURE 2

The REPLACE option is very similar, except that this time you have to give a replacement string as well as a search one and you need to tell QD whether all occurrence, or only the first are to be replaced, with or without confirmation. Once again, the process is very quick.

The SEARCH took about 3 seconds to find the single mention of the word *'zzZZzz'* in a 155 kbyte text file. The word was, of course, right at the very end and the file was, of course, the source code for DJToolkit !

QD 7 has quite a number of menus all full of commands, however, these menus have been set up in a sensible manner by grouping various related commands together on a single menu, with sub menus available where required. On the main screen there are the FILE, COMMAND, BLOCK, STATUS and WORD menus which are accessed via the mouse and/or the keyboard using the function keys.

The FILE (F2) menu pulls down and on it there are 9 options, some of which call up other menus some of which carry out an action. The FILE menu options are :

LOAD TEXT FILE .. INSERT TEXT FILE .. INSERT SCRAP (Insert text from the scrap buffer) FORGET CURRENT FILE SAVE TEXT FILE SAVE TEXT FILE AND QUIT SAVE WITH NEW FILENAME .. PRINT .. SAVE BLOCK ..

The COMMAND menu has the following 9 options :

GOTO .. MARKER .. SEARCH STRING .. REPLACE STRING .. DELETE CONTROL CODES CHARACTER TABLE .. LINE NUMBERS .. TYPING CHECK QUIT QD

The GOTO option has a couple of very useful abilities. You can select to goto an assembler label or a SuperBasic procedure or function name directly. You can CONFIG this option to sort the label/proc/function names before showing them in a list. Simply HITting or DOing the required name will take you there in the source - most useful, especially in a large program file. See figure 2 for the list of assembler labels and figure 3 for what happens when I HIT the one that is highlighted.

The typing check appears to check your speeeelling (!) as you go along provided you have QTYP2 loaded.

The BLOCK menu is only available whenever you have defined a block of text to operate upon. The following options are available :

FORGET DELETE GOTO START COPY MOVE OVERWRITE SCRAP ADD TO SCRAP UPPERCASE LOWERCASE

The STATUS menu is a bit of a misnomer as it allows you to do other things apart from getting the current status of the program. You can :

SET LEFT MARGIN SET RIGHT MARGIN SET TAB INTERVAL SET THE FILE EXTENSION SET DIRECTORY SET HELP FILE SET THE FONT USED SET THE CONFIRMATION REQUEST SET TAB OPTIONS SET BACKUP ON SAVE SET THE KEY CLICK

I did find the help file and directory useful. When you CONFIG the program, you tell it where to find the help file, however, when you come to run the program, if the help file is not there F5 followed by H will let you find the help file somewhere else and load it from there.

I found the key click, when on, to be very irritating !

The WORD menu lets you carry out certain operations on a single word. Some of the options are a bit strange, but very handy. The 9 options are :

SELECT WORD HEX TO DEC HEX TO BIN DEC TO HEX DEC TO BIN BIN TO HEX BIN TO DEC INTO HOTKEY BUFFER SET WORD DELIMITERS

Once a single word has been selected, and it could just be a number, the conversion options become available. This lets you type in, for example, a decimal number, mark it as the current word (use CTRL + W) then have QD convert it into HEX or BINARY for you. You can, of course, convert it back should you require.

QD7

QD will not convert words that are not valid decimal, hex or binary, so you couldn't select FRED as a word and convert it to decimal, for example. Well you could, but only the 'F' would be converted to 15. The converted word then replaces the original, should you confirm your desire to have this happen.

In the above, where a menu option shows a couple of dots after it, this shows that selecting that option leads to something else happening before the action is carried out. For example, on FILE LOAD, you will have to choose a file to be loaded before the load takes place. Menu options that have no dots, mean that that action will take place immediately.

All in all, I like QD 7 quite a lot. I have a few niggles (see below) about it, but most of these would probably be overcome by reading the full manual - at least I have an excuse. What I do miss, as an EDITOR SE user very regularly, is the ability to carry out commands within the editor - I do this at work on ICL mainframe editors and also in EDITOR SE and occasionally, in EMACS. QD does not seem to let you do this. Repetitive edits are also a 'missing' feature.

Now to the bad points of QD.

Bracket matching left and right allows you to position the cursor on any bracket, one of "<", "(", "{" or "[", or a single or double quote, and then press CTRL ALT SHIFT and the LEFT or RIGHT arrow to find the opening or closing bracket or quote that matches it. While I always found the matching bracket going RIGHT, QD would sometimes simply beep at me when I tried to find the bracket that I had just come from. At first this seemed to be only when going LEFT and the required bracket was off the top of the screen, but even when there were only 10 lines in the program, it still failed.

The following file gave QD the above problems.

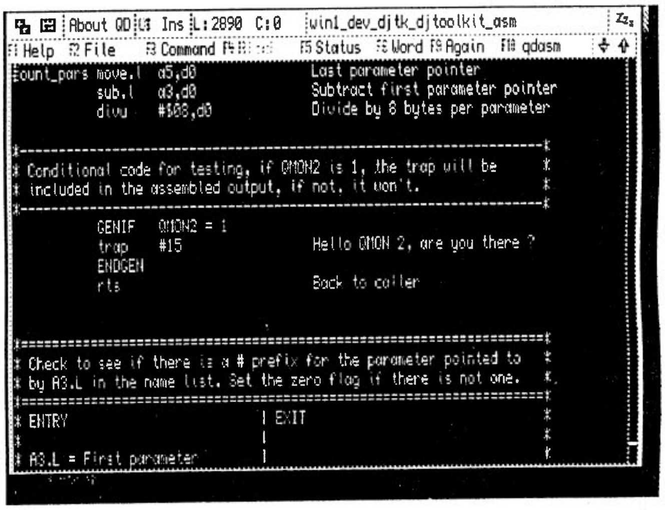
```
#include <stdio_h>

main(int     argc,     char     *argv[])     {
int x;
for  (x  =  0;  x  <=  argc;  x++)
printf("\nARGV[%d] = '%s'", x, argv[x]);

}
```

Placing the cursor on the opening brace, '{', and pressing CTRL ALT SHIFT RIGHT found the closing brace, '}'. Leaving the cursor over the closing brace and pressing CTRL ALT SHIFT LEFT did not find the opening brace that I had just come from, it simply cause QD to beep forlornly !

The algorithm used in QD to match brackets is quite good as it beeps when there is no matching bracket even if there are a number of opening and closing brackets, brackets or quotes etc between where you have the cursor and where the required bracket should be.



```
count_pars move.l  a5,d0                    Last parameter pointer
           sub.l   a3,d0                    Subtract first parameter pointer
           divu    #$08,d0                  Divide by 8 bytes per parameter

*-------------------------------------------------------------------*
* Conditional code for testing, if QMON2 is 1, the trap will be    *
* included in the assembled output, if not, it won't.              *
*-------------------------------------------------------------------*
           GENIF   QMON2 = 1
           trap    #15                      Hello QMON 2, are you there ?
           ENDGEN
           rts                              Back to caller

*===================================================================*
* Check to see if there is a # prefix for the parameter pointed to *
* by A3.L in the name list. Set the zero flag if there is not one. *
*===================================================================*
* ENTRY                          | EXIT                             *
*                                |                                  *
* A3.L = First parameter         |                                  *
```

FIGURE 3

Another problem was getting QD 7 to talk to either the QD or the QDASM thing. I tried everything to make QD 7 send an assembler source file to the QMAC assembler. It totally refused and simply appeared to trash the file that was loaded by reformatting parts of it. As I stated above, the manual might have helped. This is, however, one of the selling points for this editor. You can write SuperBasic programs (without line numbers if you wish) and then have it passed to QLiberator to be compiled. If there are errors, you get the error file loaded into QD for checking. On the other hand, you can write assembler programs in QD and have those passed to QMAC for assembly.

The program allows you to SIZE its window by CTRL + F, or MOVE its window by using CTRL + E, but why are these keys used when the 'standard' Pointer Environment keys of CTRL F3 and CTRL F4 work anyway ?

In summary. I like QD, I would have preferred to see a full manual with it - I could have given a better review then possibly. The editor is nice to use for editing straight text, it is a programmer's editor and does the job very well. It is fast and very easy to use. I miss not having a command file/repetitive command ability but maybe QD 8 will allow this. If you want a Pointer Driven editor, then this is the one for you (it is the only PE editor that I have heard about, EMACS PE doesn't count !), go out and buy a copy. If you don't need a pointer driver editor, then don't buy QD 7 especially, but bear in mind that it does allow a sort of 'Integrated Development Environment' or IDE so much loved on the PC, where editing, compiling and on the PC, testing, can all be done from within the editor.

Norman Dunbar

QD7 can be obtained from Jochen Merz Software (Germany 0203 501274), Im stillen Winkel 12, 47169 Duisburg, Germany. The price is DM 125,- plus DM 14,- P&P.

INTERRUPT

In the first part of this article on communication between different jobs, I gave an outline of a number of different ways that jobs could communicate with each other. In this part, I will give a few examples of the many ways in which the task can be carried out. As I wish to keep code to a minimum, there will be little or no error trapping and only brief comments. Where required, I will discuss the internals of the programs in the main text.

Some programs will be written in SuperBasic, some in C (using C68) and if you are lucky, some might just be in assembler. The SuperBasic ones will have to be compiled in order to use them and for this, I will use both compilers, Qliberator and Turbo. Where it is not really necessary for a SuperBasic program to be compiled, I will mention it in the text.

Just to recap on the various methods I outlined last time. We start off with the simplest, using work files to hold the communication data. Following closely behind the work files, we will have named pipes and rapidly following the named pipes there will be various filter programs.

Behind the filters there will be a large area of shared data and some things will be close behind that. The simple device list will be used next, then parent & child jobs. Similar to this, but different there will be a C68 program that causes one of its own internal functions to become a child job with access to its parent's data area and finally, Simon N Goodwin's MEM device will be put to good (?) use.

Since the first article, a new utility has appeared on some of the bulletin boards. This is called CSM and stands for Client Server Manager. The latest version seems to be 100a at the time of writing.

This utility lets programs talk to each other without explicitly using pipes etc. One program is the SERVER and this one is in control of the 'data' that other programs can request. One or more CLIENT programs send messages to the server to request 'data' and if the server can, it sends the requested data back to the client. This is proper job to job communication. I will hopefully have cracked its use by the time I get around to writing more about it!

WORK FILES

Work files are really very simple ways to pass data from one job to another. The only (?) problem is that both jobs have to know the data file name or else the whole thing won't work. On the other hand, you could, and should, write your programs so that they can take the filenames as parameters.

The following 2 small programs show how work files work. They are deliberately kept simple so that you can see the basics. There is little or no error trapping so be advised that in a 'real' program, error trapping is always required.

Although the programs are only passing a few numbers between each other, the principle works for other types of data as well. When you EXEC the compiled versions of these programs, you can have the waiting job running before the creating job. The waiting program will wait until the data file becomes available for it to read. These example use DJToolkit to trap errors. You could use DEVICE_STATUS from Turbo, FOPEN etc from Toolkit 2 or Q_ERR_ON etc from Qliberator's toolkit routines.

First we have the job that creates the work file.

```
1000 REMark Example of Job to Job communication
1005 REMark using simple work files.
1010 :
1015 REMark WORK FILE CREATOR
1020 :
1025 REMark Copyright Norman Dunbar 1995.
1030 REMark Permission given for unlimited abuse
1035 REMark of this program !
1040 :
1045 REMark First extract the work file name from our command line.
1050 IF cmd$ = ""
1055    work_name$ = 'ram1_WORK_data'
1060 ELSE
1065    work_name$ = cmd$
1070 END IF
1075 :
1080 REMark Open screen channel.
1085 s% = DJ_OPEN('scr_')
1090 PAPER #s%, 7
1095 INK #s%, 2
1100 CLS #s%
1105 PRINT #s%, 'Creating ' & work_name$\\
1110 :
1115 REMark Open OVERWRITE work file channel.
1120 c% = DJ_OPEN_OVER(work_name$)
1125 IF c% < 0
1130    PRINT #s%, 'Cannot create ' & work_name$
1135    PRINT #s%, 'Error code is '; c%
1140    STOP
1145 END IF
1150 :
1155 REMark Loop to write the data for the other job.
1160 FOR x = 1 TO 100
1165   PRINT #c%, x
1170 END FOR x
1175 :
1180 REMark Shut down & exit.
```

```
1185 CLOSE #c%
1190 CLOSE #s%
1195 STOP
```

All that the above program does is opens the appropriate work file and writes the numbers 1 to 100 to it. The file is then closed and the program exits.

Next, the job that reads the work file data created by the above example program.

```
1000 REMark Example of Job to Job communication
1005 REMark using simple work files.
1010 :
1015 REMark WORK FILE READER
1020 :
1025 REMark Copyright Norman Dunbar 1995.
1030 REMark Permission given for unlimited abuse
1035 REMark of this program !
1040 :
1045 REMark First extract the work file name from our command line.
1050 IF cmd$ = ""
1055    work_name$ = 'ram1_WORK_data'
1060 ELSE
1065    work_name$ = cmd$
1070 END IF
1075 :
1080 REMark Open screen channel.
1085 s% = DJ_OPEN('scr_')
1090 PAPER #s%, 7
1095 INK #s%, 2
1100 CLS #s%
1105 PRINT #s%, 'Waiting for ' & work_name$;
1110 :
1115 REMark Open work file channel.
1120 REPeat wait_for_work
1125    PRINT #s%,'.';: REMark So you know it is still working
1130   c% = DJ_OPEN_IN(work_name$)
1135   IF c% >= 0
1140      EXIT wait_for_work
1145   END IF
1150 END REPeat wait_for_work
1155 PRINT #s% \\
1160 :
1165 REMark Loop to read & write the data from the other job.
1170 REPeat loop
1175   IF EOF(#c%) THEN EXIT loop: END IF
1180   INPUT #c%, job_data$
1185   PRINT #s%, job_data$
1190 END REPeat loop
1195 :
1200 REMark Shut down & exit.
1205 CLOSE #c%
1210 CLOSE #s%
1215 STOP
```

This program sits in a loop until the data file that it requires becomes available and readable. When this happens, the file is opened and every line read in as a string and simply printed to the screen. At the end of the data file, the program tidies up and exits.

As you can see, both programs are very small and simple. You get the idea from the above and expand on them as your needs demand. In the above examples, the program gets the name of the work file to use from the command line when you execute the program. This requires either Toolkit 2's EX, Turbo Toolkit's EXECUTE or QLiberator's QX command to pass the parameters. The normal QL's EXEC command cannot do this although the one on the QXL can.

To run the programs, first :

```
EX 'flp1_Work_file_in_obj';'ram1_test_data'
```

will start up the program that reads in the data. It will wait in the loop until the file 'ram1_test_data' becomes available for it to read. Now start up the creation program :

```
EX 'flp1_Work_file_out_obj';'ram1_test_data'
```

Always make sure that both programs are using the same file names ! You don't have to run the programs in this order, but it shows how the reader program will wait for its data.

NAMED PIPES

In the previous article, I almost got Hans Lub's name correct, it is in fact, Lub and not Luub. I found version 1.00 of the NPIPE_REXT file on the latest release of the C68 system on the Runtime 2 disc. There is documentation on using these useful devices on the documentation 1 disc as well. What more could you need ?

In the following, I assume that you have already LRESPR'd the NPIPE_REXT file. If you have not, then it won't work !

To write a file to a named pipe, open the file as PIPE_name_size and then write to it. To read it back into another program, simply open the file as PIPE_name, where the name is the same as above.

As a small example, assume that you have a job that reads data from a file and writes it to the screen. Another

INTERRUPT

job, it could be a simple editor, creates the data that the first job will send to the screen. In the first job, open the input pipe (PIPE_name) in the first, create the data and save it to the output pipe (PIPE_name_size). The following SuperBasic program, when compiled with Qliberator or Turbo, will suffice as an example. (It uses DJToolkit)

```
1000 REMark Example of Job to Job communication
1005 REMark using Hans Lub's named pipe driver.
1010 :
1015 REMark Copyright Norman Dunbar 1995.
1020 REMark Permission given for unlimited abuse
1025 REMark of this program !
1030 :
1035 REMark First extract the pipe name from our command
line.
1040 IF cmd$ = ""
1045    pipe_name$ = 'PIPE_data'
1050 ELSE
1055    pipe_name$ = cmd$
1060 END IF
1065 :
1070 REMark Open screen channel.
1075 s% = DJ_OPEN('scr_')
1080 PAPER #s%, 7
1085 INK #s%, 2
1090 CLS #s%
1095 PRINT #s%, 'I am waiting ........'\\
1100 :
1105 REMark Open pipe channel.
1110 c% = DJ_OPEN_IN(pipe_name$)
1115 IF c% < 0
1120    PRINT #s%, 'Cannot open ' & pipe_name$
1125    PRINT #s%, 'Error is ' & c%
1130    STOP
1135 END IF
1140 :
1145 REMark Loop to read & write the data from the other
job.
1150 REPeat loop
1155   IF EOF(#c%) THEN EXIT loop: END IF
1160   INPUT #c%, job_data$
1165   PRINT #s%, job_data$
1170 END REPeat loop
1175 :
1180 REMark Shut down & exit.
1185 CLOSE #c%
1190 CLOSE #s%
1195 STOP
```

The above example program is very, very simple. All that it does is opens a screen channel and then waits for data from the pipe. The pipe may not yet have been created, but no problem, we can wait.

The pipe name used defaults to 'PIPE_DATA', but, if you want to supply a different name, simply pass it as the programs command string when EXing it.

When the pipe data becomes available, it will simply be printed to the screen channel and the program will terminate. How do you use it ?

First of all, EX, QX or EXECUTE the above (compiled) program and then at the SuperBasic command prompt, simply type this :

```
OPEN_NEW #3,'PIPE_DATA_4000'
LIST #3
```

Whatever program you happen to have loaded at the time will be listed to the named pipe channel. As soon as this data is available, the compiled version of the above program will start to list the program that you sent to channel #3. You can now close #3.

If the loaded program is very big, change the '4000' above to something bigger than the expected size of the loaded program.

(If you have the Pointer Environment loaded, you will need to PICK the compiled job in order to see its output on the screen. Any other jobs running that are using the screen will prevent it from sending its data to the screen until it becomes the top job.)

As another example, EX the above program again, but this time, load your favourite text editor (Quill won't like the file names) and save some text to a named pipe. The pipe name should be 'PIPE_DATA_nnnn' where 'nnnn' is some suitable value.

BEWARE, you can cause a bit of a lock up if the pipe being written to is not big enough to take ALL the data being written to it. I know, it happened while testing the above example.

Now what happens is that the data being 'saved' from your editor is being listed by the other job. Easy stuff.

In the next issue Filters and Client/Server programming.

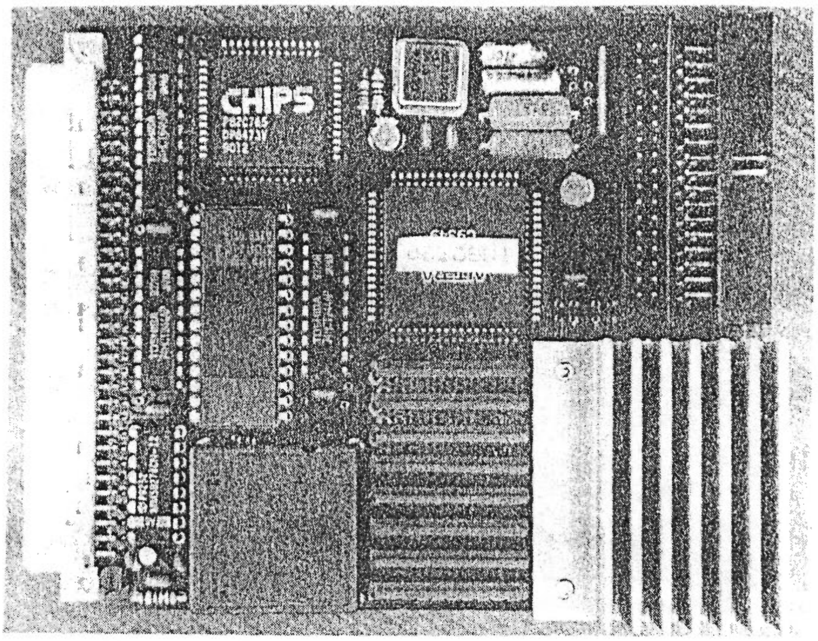
Norman Dunbar

[The programs will be available on disk for this series of articles on inter-job communication, full details will appear in the next issue.]

MIRACLE SYSTEMS

SUPER GOLD CARD

"The Pathway to Future QL Development"



Briefly...

- *3 Times Faster*
- *68020 processor*
- *4M bytes of RAM*
- *CENTRONICS port*
- *Supports 4 Disk Drives*
- *2 Year warranty*

What is it ?

The SUPER GOLD CARD is the first major revision of our highly successful Gold Card. We have replaced the 68000 processor with the 68020 so programs run about 3 times faster and have expanded the memory to 4M bytes. Additional improvments include a fast CENTRONICS printer port, 2 double disk drive ports, virtually crash-proof clock and a socket to optionally connect 5V. We also supply a 3 meter Centronics printer cable at no additional cost.

The deal...

£275 including VAT - (£240 outside EU)

(Includes postage, a 14 day money back guarantee and a 2 year warranty.)

or you can send us your GOLD CARD and:

£175 including VAT - (£150 outside EU)

You can also deduct a further £15 for a returned QL CENTRONICS and/or £10 for a DISK ADAPTER.

We are happy to accept payment by sterling cheque made payable to "MIRACLE SYSTEMS", or by quoting your MASTERCARD/VISA/SWITCH credit card number and expiry date (SWITCH card holders please also quote issue number).

Recycled Items...

Gold Card £100 (outside EU £90)   QL Centronics £ 15   Disk Adapter £ 10

(Recycled items carry a 1 year warranty.)

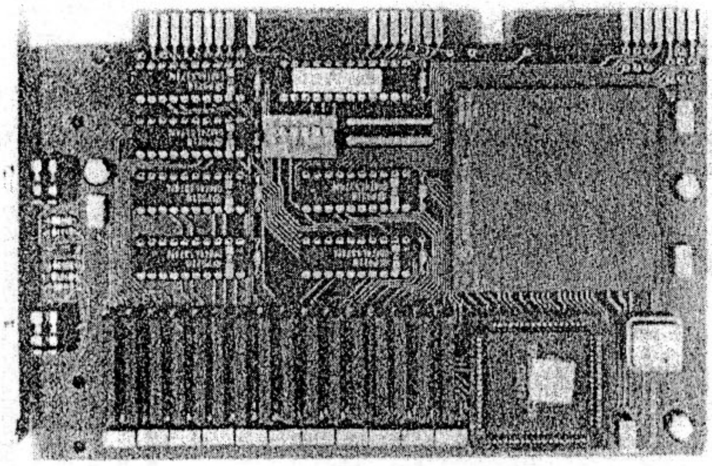
TELEPHONE/FAX: (01454) 883 602

MIRACLE SYSTEMS LTD - 20 Mow Barton, Yate, Bristol, BS17 5NF, UK

MIRACLE SYSTEMS

QXL
Now With
SBASIC

SuperBasic Compatible Interpreter



- * *68EC040*
- * *4M or 8M of RAM*
- * *Multitasking SBASIC*
- * *QL Network ports*
- * *Toolkit II*
- * *QDOS or MSDOS floppies*
- * *Uses PC's keyboard, floppy & hard disks, parallel/serial ports and mouse.*

This is the card that plugs into a standard 8 or 16 bit ISA slot on a PC and allows the PC to run QL programs - FAST. A new QDOS compatible operating system from Tony Tebby called SMSQ, which is supplied on a disk, includes Toolkit II and gives you the familiar QL environment. SMSQ includes SBASIC a multitasking SuperBasic compatible interpreter.

Installation is simple; plug the QXL into a spare slot and copy 2 files from the supplied disk onto the hard drive and you're ready to go. From the DOS prompt type QXL and the PC will transform itself into a QL before your very eyes. If at any stage you wish to return to DOS just press CTRL-ScrollLock. You can later resume the QL session by typing QXL/ which takes you back to where you left off. For POINTER ENVIRONMENT programs SMSQ can be configured to handle 3 screen resolutions in addition to the standard 512x256 QL screen. Your PC must have EGA or VGA graphics. EGA allows 640x350 whereas VGA also allows 640x480. Most SVGA cards will allow SMSQ to use 800x600 as well.

PRICING:

QXL (4M)	£280 including VAT - (£245 outside EU)
QXL (8M)	£395 including VAT - (£345 outside EU)

or you can send us your GOLD CARD and

£180 including VAT - (£155 outside EU) for a 4M - QXL

£295 including VAT - (£255 outside EU) for a 8M - QXL

You can also deduct a further £15 for a returned QL CENTRONICS and/or £10 for a DISK ADAPTER.

We are happy to accept payment by sterling cheque made payable to "MIRACLE SYSTEMS", or by quoting your MASTERCARD/VISA/SWITCH credit card number and expiry date (SWITCH card holders please also quote issue number).

TELEPHONE/FAX: (01454) 883 602

MIRACLE SYSTEMS LTD - 20 Mow Barton, Yate, Bristol, BS17 5NF,

FUNCTIONALITY

This listing is a set of useful procedures and functions which you can use in your library. Simply extract the routines you want to use in your programs and renumber as you want them. All are in structured BASIC, so there should be no problem with line numbers in GO TO statements etc.

The following notes give brief details of what each routine does. They are supplemented by REMark statements in the routines themselves.

CAPS_LOCK

Sounds a beep each time you press the caps lock key, one pitch for when you turn it on, another for when you turn it off.

DECIMAL_TO_ROMAN$

A function which converts a decimal number into roman number equivalents, only intended to work for values from 1 to 4999. It uses subtractive notation, so 9 is expressed as IX instead of VIIII, for example. It works by setting up arrays of units and reducing the decimal number to select the approriate element.

ROMAN_TO_DECIMAL%

A function which returns a decimal number corresponding to a given roman number. This will cope with either subtractive or additive notation. Note how it checks if two successive digits are greater or smaller than each other

```
6290 REMark this routine beeps when CAPS LOCK is pressed, high beep
6300 REMark for caps on, low beep for caps off, best set to very low
6310 REMark prority when compiled. Needs CAPS_LOCK_ON function above
6320 DEFine PROCedure CAPS_LOCK
6330   engaged = CAPS_LOCK_ON
6340   IF engaged : BEEP 2000,0 : ELSE BEEP 2000,50 : END IF
6350   REPeat get_key
6360     REMark wait until it changes
6370     key = CAPS_LOCK_ON
6380     IF key = engaged : NEXT get_key
6390     engaged = key
6400     IF engaged : BEEP 2000,0 : ELSE BEEP 2000,50 : END IF
6410   END REPeat get_key
6420 END DEFine CAPS_LOCK
6430 :
6440 DEFine FuNction DECIMAL_TO_ROMAN$(de)
6450   LOCal hun$(9,4),ten$(9,4),one$(9,4)
6460   RESTORE
6470   DATA 'C','X','I','CC','XX','II'
6480   DATA 'CCC','XXX','III','CD','XL','IV'
6490   DATA 'D','L','V','DC','LX','VI'
6500   DATA 'DCC','LXX','VII','DCCC','LXXX'
6510   DATA 'VIII','CM','XC','IX'
6520   FOR num = 1 TO 9 : READ hun$(num),ten$(num),one$(num)
6530    RETurn FILL$('M',de DIV 1000)&hun$((de MOD 1000)DIV 100)&ten$((de MOD 100)DIV
10)&one$(de MOD 10)
6540 END DEFine DECIMAL_TO_ROMAN$
6550 :
6560 DEFine FuNction ROMAN_TO_DECIMAL%(rom$)
6570   LOCal roman$(7,2),decim%(7),conv,total%,subscript%,first%,second%
6580   REMark LOCal r,num1$,num2$
6590   REMark too many variables to be local on old roms, add other locals
6600   REMark if you have Minerva, for example, by removing REMark
6610   roman$(1) = 'M' : decim%(1) = 1000
6620   roman$(2) = 'D' : decim%(2) = 500
6630   roman$(3) = 'C' : decim%(3) = 100
6640   roman$(4) = 'L' : decim%(4) = 50
6650   roman$(5) = 'X' : decim%(5) = 10
6660   roman$(6) = 'V' : decim%(6) = 5
6670   roman$(7) = 'I' : decim%(7) = 1
6680   total% = 0 : subscript% = 1
6690   REPeat conv
6700     IF subscript% > LEN(rom$) : EXIT conv
6710     num1$ = rom$(subscript%) : first% = 0
6720     FOR r = 1 TO 7
6730       IF num1$ == roman$(r) THEN
6740         first% = decim%(r)
6750         EXIT r
6760       END IF
6770     END FOR r
6780     subscript% = subscript% + 1
6790     IF subscript% > LEN(rom$) : total% = total% + first% : EXIT conv
6800     num2$ = rom$(subscript%) : second% = 0
6810     FOR r = 1 TO 7
6820       IF num2$ == roman$(r) THEN
```

FUNCTIONALITY

to decide whether to use subtractive or additive methods (e.g. subtractive is where you have a number such as IX, where the I has to be subtracted from the larger value X following it.

I hope you find these routines useful - if you find any bugs, please let me know so that I can get a correction printed.

Dilwyn Jones

```
6830         second% = decim%(r)
6840         EXIT r
6850       END IF
6860     END FOR r
6870     IF first% >= second% THEN
6880       total% = total% + first%
6890     ELSE
6900       total% = total% + second% - first%
6910       subscript% = subscript% + 1
6920     END IF
6930   END REPeat conv
6940   RETurn total%
6950 END DEFine ROMAN_TO_DECIMAL%
```

PICTURES AT A WORKSHOP

THE QUANTA TEAM

Bill Newell **Phil Jones** **Roy Brereton**

HARDDISKS & SOFTWARE

If you have just bought a hard disk, you may have a lot of files on floppy disk you want to copy to directories on hard disk. So to copy all of the Quill _DOC files into the directory WIN1_QUILL_LETTERS_, here is all we have to do:

```
WCOPY FLP1_ _doc TO win1_Quill_letters_ _doc
```

Notice that it doesn't really matter if the names are in upper case. The commands work out what is either side of the double '_' and do their best to match things up accordingly. You have to be careful about typing the correct names, and being careful with single and double '_' characters and so on.

Note that strictly speaking, FLP1_ or FLP2_ although being drive names, are in some ways directories themselves. In fact, they can be regarded as being the 'root' directory from which the 'tree' of filenames grows.

See figure 1 for a pictorial representation of a directory tree, with the lines showing from which part another part grows. Try to match up the contents of the diagram with the examples I listed above. Once you have started to create directories and put files in them, try to draw your own 'tree' diagram like this. Until you get used to remembering the names, they may actually be a help to you to locating your files!

So far so good, but these toolkit 2 commands only operate on the content of one directory at a time. Supposing we have Quill _doc files all over the place - for example, many commercial programs seem to have a file called something like UPDATES_DOC to tell you about changes in the latest version of the programs. How on earth are we supposed to find all these scattered in various directories on your hard disk - you don't have a hope. Or do you? What we need is a command to look through every directory on a given drive and list all files with a given ending. Phil Borman thought of this and provided us with the 'WDD' commands in the file called WD_BIN supplied with Qubide.

If we enter the command WDDIR WIN1_ _DOC, it will take a few seconds to look through all directories on the drive to find each and every file which has a name ending with '_doc'. If you have a large number of such files and perhaps a large number of directories it will take a while, but will get there. Similarly, WDSTAT will act like WSTAT in Toolkit 2 but not just look in the directory you specify, in all the directories within that as well. So for example, WDSTAT WIN1_ would list ALL files on that drive, no matter where they were hiding (it may take a while though).

The other two are WDDEL, which will delete all files in a directory and also files in any directories also inside that directory. Ouch, I hear you say, it could be very dangerous, and you'd be right. It does have a use, however. Suppose you want to remove a directory called WIN1_FRED_ altogether, because it contains obsolete files. If you tried DELETE WIN1_FRED_ it might well give the error report 'in use'.

OK, so you go to delete files one by one, until you remember that FRED also contains directories called JIM and JOE (WIN1_FRED_JIM_ and WIN1_FRED_JOE_), which in turn contain more files and so on. It would be a nightmare going through these finding every single one. So instead we use the command WDDEL WIN1_FRED_ and every single file (except for the directory names themselves, which are easy enough to remove by hand) will be deleted after a confirmation from you. WDDEL is bad enough, but at least it asks for confirmation before doing anything drastic. WDDEL_F is even worse, it just goes ahead and deletes everything without even asking. Thank you very much, just what I've always wanted!!! And usually you would make that mistake just before making a backup as well......oh dear!

TREE

There is a very useful extension to list the directory tree (i.e. all the directories and subdirectories). The command TREE WIN1_ will print to the screen a list of the directory names (without the filenames they contain). Very useful when you start to accumulate a lot of directories on your drive. It produces a list like this:

```
QUILL
QUILL_LETTERS
QUILL_INVOICES
FRED
FRED_JIM
FRED_JOE
```

and so on. So you know the sub directory called LETTERS is inside Quill's own directory. In simple setups, you will not use this much, but if like me you have a large capacity hard disk, this starts to become an essential command!

Incidentally, many of these extensions can have an extra channel number to direct the listing to a window on the screen or even to the printer:

```
TREE #1,FLP1_ : REMark lists to the screen

OPEN#3,SER1:TREE #3,WIN1_:CLOSE#3 : REMark to the printer
```

figure 1 - the example directory tree, alternative view to help to visualise a directory tree. Root is WIN1_

```
=========================================================

WIN1_
  |
  |--QUILL
  |     |--LETTERS          this is WIN1_QUILL_LETTERS_
  |     |
  |     |--INVOICES
  |     |
  |     |--FRIENDS
  |
  |--BASIC
  |
  |--TEXT87
  |     |--DOCS             this is WIN1_TEXT87_DOCS_
  |     |
  |     |--FONTS
  |     |
  |     |--DRIVERS
  |
  |--CLIPART
  |     |
  |     |--EYEQ
  |     |
  |     |--PAINTER
  |     |
  |     |--GIFS
  |     |     |--COLOUR     this is WIN1_CLIPART_GIFS_COLOUR_
  |     |     |
  |     |     |--MONO
and so on!
=========================================================
```

Incidentally, a little tip gained from hard experience. Most of the programs which can be started without the fuss of loading extensions and so on are best grouped in one directory. I put them in a directory called WIN1_EXEC_. This hasn't stopped me wasting space by putting every file which belongs to it elsewhere, but when I need a program quickly, I know where it's likely to be. So instead of trying to remember where I put a program called PCX_obj six months ago, I know there's likely to be a copy of it in WIN1_EXEC_. Of course, if you prefer not to do this, you can always use commands like WDDIR to locate them.

SUB - THE SUBSTITUTER

All this is all well and good with more modern well written software, but many programs simply can't be told to run from WIN1_EXEC_ or wherever you put them. Take Quill as a simple example. You put it in a directory called WIN1_QUILL (so it is called WIN1_QUILL_QUILL, an unfortunate choice of names perhaps). It is easy enough to start it from there with the command EXEC_W WIN1_QUILL_QUILL. But what happens when it needs to find its PRINTER_DAT file? From using your configuration program supplied with Quill, you may remember that you can only tell it to look on MDV1_, MDV2_, FLP1_ or FLP2_. It might just be possible to cheat and hack or tinker with the code a bit to make it look on WIN1_, but a five character name is all it will handle. It's admittedly not too much of a problem to save a copy of the file called WIN1_PRINTER_dat if it is the only awkward program you have, but that stops Archive having its own copy of PRINTER_DAT, for example. So it would be nice if Quill would like in its own directory.

With some software, we could set the Toolkit 2 default to the directory name and some programs would look there:

```
DATA_USE WIN1_QUILL_:EXEC_W WIN1_QUILL_QUILL
```

Unfortunately, software like Quill ignore this default drive and insist on searching on FLP1_, getting a bit upset when it doesn't find the PRINTER_DAT driver file and proceeds to print your file wrongly. Grrr. Time to rethink all this. So let's do some head scratching and see what we come up with.

No need, because good old Uncle Phil (the nice chap who put together the software for the Qubide interface) has done some head scratching for us already and come up with the SUB facility. SUB stands for substitute or sub-directory and it is effectively a cheat mechanism for stubborn software like Quill.

What it does is to allow you to say that Quill should look in a long directory name, but it fools Quill into looking in its usual short drive name. Effectively, what it does is to magically translate the 'FLP1_' which Quill thinks it's looking on into another longer directory name. Let us assume we have an imaginary drive called SUB1_. We know our hard disk is called WIN1_, our floppy drive's called FLP1_ and so on. You can define SUB1_ to be whatever you want (as long as it's legal, decent and honest). First, choose which drive to use with the SUB_DRV command:

```
SUB_DRV WIN1
```

Now choose the sub-directory names. We'll use WIN1_Quill here for the first device, as Quill is normally set up to look for its system files on drive 1:

```
SUB_USE 1,QUILL
```

What we have now done is to make the imaginary drive called SUB1_ actually pretend to be WIN1_QUILL_. So when we type in the command DIR SUB1_ it gives a list of files in the directory called QUILL on WIN1_. In other words, it's the same as the command DIR WIN1_QUILL_, but the drive name is only five characters long. Some software will only allow a five character name, but will allow that to be any 5 character name, so not objecting to the drives being called SUB1_ and so on. Quill would let you do this, but you'd have to do a bit of hacking in the code to make sure that all references to FLP1_ were changed to SUB1_, then save this modified copy of Quill onto the hard disk once more, and remember to set up SUB1_ each time you use Quill.

We can take this a step further in fact. We can define more SUB drives for our various file directories. For example, SUB2_ could be disguised as WIN1_QUILL_LETTERS_, SUB3_ could be WIN1_QUILL_INVOICES_ and so on. The limit is 8 devices. So we might end up with something like this:

```
SUB_DRV win1
SUB_USE 1,QUILL
SUB_USE 2,QUILL_LETTERS
SUB_USE 3,QUILL_INVOICES
```

Oops, we have forgotten already what we set the SUB names to. Phil Borman thought of this too (isn't he a clever chap?). He has provided a function called SUB_USE$ to remind us what each SUB device was called. So PRINT SUB_USE$(2) would tell us SUB2_ was actually looking in a directory called QUILL_LETTERS. Another function called SUB_DRV$ tells us that it was working with WIN1_.

You will have noticed that we used a number after SUB_USE in the example above. There is another facility to change the name of the SUB device itself for use with programs which won't even allow names like SUB1_. Suppose that the program concerned allowed the use of a name such as FDK1_ as well as FLP1_. We can now make the SUB device respond to the name FDK instead of SUB and fooling that program completely. So we use the version of SUB_USE without a number to change the name of the SUB device.

```
SUB_USE fdk
```

So what used to be called SUB1_ no longer exists. It is now called FDK1_. DIR fdk1_ would produce the same result as DIR SUB1_ used to, except that the floppy disk drives called FDK1_ no longer work. Aaargh!!!!

Most floppy disk interfaces include a command called FLP_USE to change the name of their device drivers. So if your drives are normally called FDK1_, you could change them to something else, say FDV1_, with the command FLP_USE fdv. This is starting to get really complicated. It is, even for experienced users like me. What we are doing is renaming all our drives to suit the software, or rather to fool the software, though this process can end up fooling us as users too! It is a difficult thing to work out, but once done correctly it works well. To make Quill work with the above examples, we'd use FLP in place of FDK, but the principle is the same. Here is an example of a simple little basic program to be run before executing quill.

```
100 REMark set up SUBs for Quill.
110 FLP_USE fdv : REMark allows floppy disks to be used
still
120 SUB_DRV win1 : REMark look in directories on win1_
130 SUB_USE 1,quill : REMark subdirectory names
140 SUB_USE 2,quill_letters
150 SUB_USE 3,quill_invoices
160 SUB_USE 4,quill_friends
170 SUB_USE flp : REMark now called FLP1_ in place of
SUB1_ etc
```

Now all you have to do is EXEC FLP1_QUILL or EXEC WIN1_QUILL_QUILL (your choice, the name win1_ is still there, but the renamed sub device is shorter). Now when printing, Quill looks on FLP1_ for its printer driver as it used to, but in reality FLP1_ is now WIN1_QUILL_. When saving Quill doc files, to save them to hard disk in the letters directory, save to FLP2_. To save invoices, save to FLP3_, and to save letters to friends save to FLP4_. Quill allows you to type in any valid 5 character drive name, it is when it looks for its own files like PRINTER_dat that the problems really starts.

Oh, what happens if you want to save files direct to floppy disk? Any attempt to save to FLP1_, or FLP2_ etc would result in the file being put on hard disk against your wishes. This is the purpose of the FLP_USE fdv statement, which gives the floppy drives another name, in this case FDV1_ or FDV2_. So instead of typing in FLP2_TEST_DOC to save it onto the hard disk, you'd enter FDV2_TEST_DOC to save to floppy disk instead.

You have every right to be confused by now. If you aren't, I am. While typing this into Quill I have been using all sorts of device names, renamed names and so on and I don't know where I am by now. I have just tried to save this article and oops, it's disappeared. In other words, it ended up somewhere other than where I tried to save it, probably because I have renamed drives over and over again and tried to be too clever. So I need to undo things quickly in case the QL crashes or some such disaster, and here is what I do:

```
SUB_USE sub:FLP_USE flp
```

That gives me control of my machine back long enough to save a copy of the file to floppy disk until I know what I've done wrong.

Finally then, a summary of the questions to ask yourself when deciding whether and how to use SUB.

(a) Decide on directories to be used.

(b) Copy the files in there (but note, some software may demand that you reconfigure before doing this, usually older programs)

(c) Configure it for hard disk, but is this possible? Will it accept a drive name such as WIN1_DIRECTORY_ ? Usually, modern software such as pointer driven programs will run from hard disk without any change other than to specify a drive and directory name. Some programs only have one choice of directory names built in, i.e. the program will run from one named directory and nothing else, so bad luck if you have two programs needing the same directory name!

(d) If it won't configure for a long enough name for a drive and directory, will it allow a standard 5 character drive name such as SUB1_ (or any name you care to use which doesn't prevent you using your disk drives, for example). If so, try to make it respond to SUB1_, SUB2_ etc and have a short basic program in its directory to set these, it might be as simple as this, and you can save one for each program using a simple standard filename such as SET_DEVICES_bas:-

```
    100 SUB_USE sub:SUB_DRV win1
    110 SUB_USE 1,DIRECTORY1:SUB_USE 2,DIRECTORY2
    120 EXEC SUB1_PROGRAM_NAME
```

(e) If like Quill there is no way to change the programmed devices in the program, you'll have to go through the whole process of setting up the full emulation names.

What device name(s) does the program insist on using?
  FLP1_ for program files, FLP2_ for data and other files

Does that clash with disk drives or anything else? YES

What can I change the names of those to instead? FDV
  FLP_USE fdv

Where have I put the program? WIN1_prog_
  SUB_DRV win1

How many SUB devices do I need? 1 and 2
  SUB_USE 1,prog_
  SUB_USE 2,prog_

What do I need to call SUB instead of SUB1_ and SUB2_? FLP
  SUB_USE FLP

I'm sure it would be possible to write a little program which itself asks these questions and creates a short basic program to sort all this out for you! Remember that basic programs are stored as plain text on a disk, so you can make this 'program generator' print suitable line numbers and commands and names to disk to create a basic program.

This article is not the easiest to follow, so if you don't understand it first time, don't be too surprised. Read it again, and try some examples for yourself, but be careful only to use unimportant files as it is very easy to cause a disaster while experimenting with fairly drastic commands like these. I didn't understand it fully at first and made several mistakes until I finally mastered it. Even now, I can't remember all of the commands and I certainly can't make all programs work at the first attempt, if they are stubborn enough to need the use of SUB, they will often need a bit of trial and error.

I have developed a program called DEV MANAGER, available from Quo Vadis Design, to make life easier when using stubborn programs from hard disk. It allows you to build up lists of device names, defaults, DEV and SUB names etc for all your software (well, not all, some old crude protected software will prove stubborn still) and then to store these lists on disk. All you do is start the program, load the list and choose a program name. It then sets all the devices for you before starting the program.

Dilwyn Jones

S.J.P.D.

Est. 1991.

36, Eldwick Street,
Burnley,
Lancashire,
England.
BB10 3DZ.
Tel. [UK] 01282 451854.

ADVENTURE 93. (2 DISKS)
STAR TREK DISKS (2 DISKS)
AMATEUR RADIO DISK.
ESCHER GRAPHICS DISK.
** TEXT 87 DEMO DISK.*
ERGON DEVELOPMENTS DEMO.
LINEDESIGN CLIP-ART. (3DISKS.)
** QL HACKERS JOURNAL.*
QeM COMMS PACKAGE.
QFAX.
** Q.T.P.I. POINTER COMMS PACKAGE.*
CROSSWORD DESIGNER.
PAGE DESIGNER 3 DEMO.
INTERNET GUIDE TEXT FILES.
BISON VERSION 1.22.
FLEX VERSION 2.4.6.
** INTERCAL.*
** UNGIF RELEASE 91.*
** 7 DISKS OF RELIGIOUS CLIP-ART.*
** GNU DIFF.*
** QL BBS READER.*
** C68 RELEASE 4.14c.*
** LIBCURSES RELEASE 1.23.*
** QPTR COMPANION. V3.04.*
** GNU TEXT UTILITIES. V1.9.*
** REVISION CONTROL SYSTEM.*
**XCHANGE VERSION 3.90J.*
** SCHEME INTERPRETER.*
** 20+ DISKS LINEDESIGN CLIP-ART.*

* = NEW OR UPDATED DISKS.

THIS IS A SMALL SAMPLE OF THE
100 + PACKAGES ON OFFER.

I CAN NOW OFFER OVER 85 DISKS OF CLASSIC LITERATURE. THESE INCLUDES THE WORKS OF SHAKESPEARE, HOMER, LEWIS CARROLL & MILTON.

AMIGA QDOS EMULATOR.
RELEASE 3.21.
ONLY £3.50p.
FULLY INCLUSIVE.

PLUS 51 DISKS OF THE BEST ASSORTED QL PUBLIC DOMAIN & SHAREWARE

THE FOLLOWING PC SHAREWARE TITLES ARE AVAILABLE ON 3.5" PRE-COPIED DISKS.

Z80 V3.02. SPECTRUM EMULATOR.
XTENDER. ZX81 EMULATOR.
APPLE][EMULATOR.
C64 EMULATOR.
QDOS READER.
ATARI 400/800 EMULATOR.
DRAGON 32 EMULATOR
AMSTRAD CPC EMULATOR.
SHOWQS.
PSION 3 EMULATOR.
MACHINTOSH EMULATOR. £5.00.
600+ SPECTRUM GAMES. £20
MOST PC DISKS ARE £1.75p
FULLY INCLUSIVE.

QL Pre-Copied Disks Are £1.75p Each, Inclusive Of Media And Return Postage.
I Can Also Copy Onto User Supplied Disks At £1.00p Per Disk Plus Return Postage.

For A Catalogue / Sampler Disk Which Details All PD & Shareware Offered, Along With Second-User Hardware, Software, Magazines & Books For Sale. Send A Formatted Disk And Return Postage Or 4 X First Class Stamps.

A Printed List Is Also Available, Send Large S.A.E. Along With 4 X First Class Stamps.

THE FORTH DIMENSION

It was the QL that introduced me to FORTH. It was the strength of the QDOS+FORTH axis that kept me loyal to the QL. And now that I have added a QXL to my two QLs I find that SMSQ and FORTH make every bit as good a pairing.

The following is not intended as any sort of introduction to FORTH, and only those of its features needed for the examples will be explained. For a more complete oversight of the language I would refer readers to a pair of articles by Charles Gerrard in SQLW for August and September 1986. The best summary I know is to be found in the documentation provided with Digital Precision's v2.0 implementation of FORTH-83, which was written by Gerry Jackson. This is an excellent piece of software which I have been using for over eight years now. I keep trying out other versions, and on other machines too, but have so far always come back to the QL and DP's with relief. It is remarkable in that after all this time I have only hit two definite errors - which were easily patched out, one of FORTH's virtues being the facility with which this can be done.

If at any point I appear to make a statement whose validity is restricted to a particular FORTH, it will be to the DP version that reference is being made.

FORTH's origins go back to the early sixties when memory, "core", was limited, and I/O, typically by paper tape punch, painfully slow. There was much to be gained from a programming method which economised on these resources. I refer to a "method" since FORTH does not possess all the characteristics which one usually expects to find in a language. There are no analogues of statements, and therefore of assignments, nor of syntax. All it has in common with, say, ALGOL or BASIC is structure, and it has that to an extent which should satisfy the most exacting of the school persons among us. There are not two separate entities : a "program" which is written in a "language". Anything that one writes becomes an extension to FORTH, however temporary. The interpreter makes no distinction between a definition present in the dictionary as part of the original package, and one that I have newly entered from the keyboard. Except that mine is the more likely to hang the machine.

FORTH resists classification, but it is best thought of as an interpreted language. Each item encountered by the interpreter in its data stream represents a complete instruction which is not sensitive to context. Thus if BASIC were to meet the group '123' the particular meaning would depend on where it found the thing. If it was at the start of a line then it would be taken to be a line number. If it was embedded in a line it would be part of a statement; to the right of an equal sign it must be a constant forming a term in an expression; to the left it would be a syntax error. Whereas to FORTH it always has the same meaning : " evaluate the string integer '123' using the current base, and place the result at the top of the data stack ".

Instructions are termed "words", a perennial source of confusion, and such words may be arranged without restriction in any order which may seem relevant to the programmer. Which might seem strange in view of the previous statement about structure. It is possible to conceive of a BASIC routine which was written entirely as a list of procedure names. Think of that, and then add the restriction that all structural statements (IF...END IF : FOR...END FOR) should be confined within the definitions of those procedures, and you have something very like a FORTH program.

A big difference between the two is that while it is a BASIC requirement that all the procedures should be predefined or have their definitions present somewhere in the program, the FORTH interpreter insists that it should only be faced with words that are already there in the linked list, or "dictionary", in which they are stored.

Words may be defined either as machine code primitives, or as structures compiled from the previously defined words in the high level language. In the former case the code is loaded directly into the dictionary according to the user's instructions on which no check is made; in the latter the rules appropriate to the particular structures employed must be resolved within the word or compilation will fail.

Two features of FORTH which have been retained from its early days are now, at least in my opinion, redundant. I feel that it would gain a wider acceptance if they were dropped.

Every time the interpreter reads the name of a word it searches the dictionary for a match. If it does not find this then it assumes the name to be really a number and tries to evaluate it. If this fails then the error routine is called. With 8-bit processors searching slow memory this could be painful, and an entity called a "Vocabulary" was introduced which was a subset of the main linked list. By compiling words into such a vocabulary, and specifying that they should be searched first in particular applications, considerable gains in interpretation times were achieved. With faster processors and memory the need for this feature has virtually vanished. The ability to manipulate vocabularies is, however, a symbol of virility that requires display. So they continue to be used on occasions when they may actually impede progress rather than assist it. But they must be seen to have been deployed.

Another way of speeding things up, and which is no longer necessary, was to divide programs up into blocks, usually of 1k, which were read as chunks into a dedicated area of memory. Within these blocks there are no line terminators, although most FORTHs are provided with specialist editors which display them as if there were, and the interpreter can whizz through them. But QDOS works with slave blocks, and it seems absurd to have one's system reading bytes from non-volatile memory into its own blocks, then solemnly transferring them across into FORTH's buffer. I ditched the block method very early in my usage of FORTH, and restricted my input to terminal lines, sourced either from the keyboard or from conventional files. Yet one still finds the aficionados publishing their work in blocks. Very tiresome it can be to work through them, too, since they tend to impose a quite spurious structure on a program which is often at odds with reality.

However the block buffer remains, protected from invasion by the compiler/interpreter if not by the programmer. I use it extensively as a general purpose scratch pad.

FORTH-83 is an integer FORTH. Addresses of locations within the dictionary are specified as twos complement integer displacements from a base address. This means that the size of the dictionary is limited to 64k. However words are provided which allow access to the full double integer addressing range. Since these take one outside the standard FORTH care has to be taken in their use. To take advantage of this external addressing facility I did a big RESPR() when booting my FORTH-dedicated QL, and then EXECed the job which ran the FORTH proper. The size of the reserved area was judged so that there was just enough space under the job header to allow QDOS to function. It was contiguous with the top of the FORTH which allowed a seamless transition between integer addresses within the dictionary and double without. I wrote traps into my own programs to hinder, if not completely prevent, me from straying outside the enlarged area.

Things are different now that the QXL has become my FORTH machine. SMSQ appears to carve all RESPR()s out of the common heap (as MINERVA does its secondary dittos). I therefore add the extra memory required to the data space specified in the file header which has exactly the same practical effect as before. This is a doddle in SMSQ as I keep the job file in WIN1_, which is itself a MSDOS file. Thus the job file header may be edited by the use of a suitable PC utility.

One of the words that may be encountered by the interpreter is a compiler which adds new entries to the dictionary. Its function is to write the following into the next available locations in memory :

2 bytes	Link field to last entry
1 byte	Length of the name string
n bytes	The identifying name itself
[1 byte	Padding to even address as required]
2 bytes	Code pointer
2n bytes	The parameter field

The address of the code pointer is termed the compilation address, and compiler defined words are simply lists of such addresses. The content of the code pointer is the address where the interpreter will find its next instructions.

If the name is that of a constant then the interpreter will be told to push the number contained in the bytes following the pointer on to the top of the data stack.

If a variable then the compilation address plus two is pushed on to the stack, the appropriate number of bytes having been reserved for the particular size of variable.

For a compiled word the instructions will push the current value of the interpretive pointer on to the return stack and then branch to the compilation address plus two. This process will proceed through successive addresses in the parameter field until that of the word "EXIT" is reached which returns the interpreter to the previous level.

For a machine code word the pointer will simply point to the first instruction word, and the interpreter will just keep on processing. It is the user's responsibility to write the preamble into the dictionary, and to include at the end of his code the necessary "NEXT" sequence which restarts conventional interpretation.

Given all this, let's try a few examples to show the consequences.

Too be continued in the next issue ....

P Tanner

Do YOU Require the very latest information on software and hardware developments within the International QDOS/SMSQ Community ? ***Then you need a subscribtion to:***

IQLR
International QL Report

Do YOU Want to Know what QLer's really think about the products they buy and do they live up to the suppliers claims? ***Then you need a subscription to:***

IQLR
International QL Report

Do YOU Need up-to-date information on show events, hints and tips, who is doing what and why, communications, PD and Shareware software developments and plenty of adverts to help you make that important decision? ***Then you need a subscription to:***

IQLR
International QL Report

Is it Important to YOU to have your QL magazine delivered to you ON TIME, EVERY TIME, without ever missing an issue? ***Then you need a subscription to:***

IQLR
International QL Report

IQLR

HOW to SUBSCRIBE: Send the appropriate amount listed below in the form of a Postal, Bank, Euro or Personal cheque to our North American or European office.

IQLR
International QL Report

Please Note: Personal cheques in £ MUST be drawn on a UK bank. Personal cheques in US$ MUST be drawn on a US bank. Please make ALL cheques payable to IQLR.

IQLR
International QL Report

Credit Card Users may subscribe by either calling or sending their Credit Card number and Expiry date to: MIRACLE SYSTEMS LTD - 20 Mow Barton - Yate, Bristol BS17 5NF - UK   Telephone/Fax: (01454) 883 602

IQLR
International QL Report

SUBSCRIPTION RATES: To the US the rate is $24.00 - To the British Isles and Europe the rate is $34.00 or £25.00 or DM70 - To Canada the rate is $27.00 - To Central and South America the rate is $34.00 - To the Rest of the World the rate is $40.00 or £30.00 or DM80.

IQLR
International QL Report

IQLR • P.O. Box 3991, Newport, RI 02840-0987 • USA
Telephone/Fax: +1 401 849 3805
IQLR • 23 Ben Culey Drive, Thetford, Norfolk • Great Britain IP24 1QJ

sinclair Z88

OFFICE/FAX 01494-871319 (EEC) W.N. Richardson & Co.

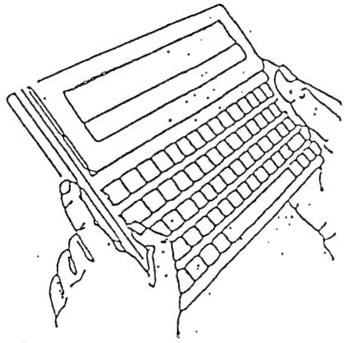
MOBILE: 0850 597650

6 Ravensmead
Chalfont-St-Peter
Buckinghamshire, SL9 0NB

THE IDEAL PORTABLE COMPANION FOR THE QL

THE CAMBRIDGE Z88 A4 NOTEBOOK

WITH BUILT-IN WORD PROCESSOR, SPREADSHEET, DATABASE, BASIC, CALCULATOR, CLOCK, ALARM, CALENDAR, & VT52 TERMINAL.
USES 4XAA ALKALINE CELLS(c.20 HOURS)



We have acquired a limited stock of the versatile

CAMBRIDGE Z88 A4 NOTEBOOK COMPUTER & ACCESSORIES.

UNTIL FURTHER NOTICE, THESE ITEMS ARE ON SALE AT SPECIAL BARGAIN PRICES. INCLUDESVAT

ITEMS.	BARGAIN PRICE.	ORIGINAL PRICE.
CAMBRIDGE Z88 (UK) COMPUTER.	£99	£189.95
32k RAMPACK	£16	£15.95
128k RAMPACK	£28	£32.95
512k RAMPACK	£60	£87.95
1Meg RAMPACK	£120	£186.95
32k EPROM PACK	£16	£16.95
128k EPROM PACK	£22	£23.95
256k EPROM PACK	£45	£57.95
EPROM ERASER	£32	£38.95
PARALLEL PRINTER LEAD	£22	£18.95
SERIAL PRINTER LEAD	£18	-
MODEM	£90	£114.95
MAINS ADAPTER	£15	£9.95
TOPPER (PROTECTIVE COVER)	£12	-
CARRYING CASE	£16	-
TRANSFER TO OTHER COMPUTERS		
SPECIAL Z88-QL SER LEAD	£10	-
COPY DISKS OF QUANTA PROGS IMP/EXP & ARCHIVE EXPORT	Each £2	-
PCLINK 5.25 KIT	£25	£27.95
Z88 TO MAC. KIT	£30	£32.95

OUTSIDE EEC. LESS 10%

AGENTS WANTED FOR ALL COUNTRIES.

POSTAGE UK £5. EEC £15. USA £20. OTHER COUNTRIES £30.

THE AVAILIABILITY OF STOCK VARIES, SO THE BARGAIN PRICES ARE ONLY OFFERED WHILE EXISTING STOCKS LAST.
ALL THE STOCK IS NEW AND IS WARRANTIED FOR 90 DAYS. IN THE EVENT OF REPLACEMENT BEING AGREED, BUT THE ITEM BEING OUT OF STOCK AT THE TIME, A REFUND WILL BE MADE PROVIDED THE ITEM IS RECEIVED IN GOOD CONDITION.

QL & PC COMPUTER USERS WILL FIND THE CAMBRIDGE Z88 ESPECIALLY USEFUL FOR WORK AWAY FROM THE DESKTOP. WITH TRANSFER PROGS DATA CAN BE SAFELY EXCHANGED WITH THEIR DESKTOP SYSTEM.

W.N. RICHARDSON & CO CONTINUES TO PROVIDE FULL SPARES AND SERVICES FOR THE SINCLAIR QL COMPUTER & THE CAMBRIDGE Z88.

JAN 95

PUBLIC DOMAIN

Dear fellow QL Public Domain & Shareware users.

Well it been another busy time up here at ***S.J.P.D. SOFTWare.*** In the last weeks I have converted some 20 disks of **clip-art** into Adobe Illustrator format suitable for use with LineDesign. These are listed on the current catalogue disk. I still have 1000's of images in PC format awaiting conversion. I will be adding these to the library over the next few months.

I have also added the following disk to the **SJPD** section:-

SJPD 49.

Qpoint Demo. A Demo Version Of This Commercial BBS Point Program.
Doc View. A Small Pointer Driven Quill Doc Viewer.
Lfax V5.0. The Latest Version Of The QL Fax Program.
Engif 0.16. A Small Utility For Converting QL Screens Into GIF Format Screens.
Rogue. An Ascii Adventure Game.

SJPD 50.

Norback 4.01. The Updated Version Of This Hard-Disk Back-Up Suite Of Programs. Requires The Pointer Environment And Menu_Rext To Work
Solvit Plus Demo. This Is The Demo Version Of A Word Game, Anagram Solver & Dictionary. The Full Version Is Available From Dilwyn Jones.
Signal. This Is The First Port Of The Signal Extensions To The QL.
LineDesign Clip-Art. A Small Collection Of 14 _AI Clips For LineDesign.
QL Faq's The Full Collection Of Frequently Asked Questions About The QL.

SJPD 51.

Unarj This Is The Executable And Source For This Popular Compression Utility As Is Common On The PC.
Env 1.07. The Latest Version Of This Environment Utility. NOW SMS COMPATIBLE.
Norback 3.23. An Updated Version Of This Hard-Disk Back-Up Suite Of Programs. FOR NON POINTER USERS.
C68 Bug Fixes. A Collection Of Bug Fixes For Version For C68 By Erling Jacobsen.
Smash. This Is A 'Self Extracting Job Compressor For QDos Compatible Systems.'
Atari Keyboard Editor. This Is Used To Edit Atari Keyboard Layout, For Use By QDos Emulator Users.
Jpeg To Gif. A Suite Of Utilities For Converting JPEG Pics. To GIF Format.
Client Server Manager. This Utility Is A 'THING' That Implements Easy And Robust Messaging Between Jobs.

The **SJS** section of the library has also grown with the additions of the following disks:-

SJS 103:- **GNU DIFF.** A New Port Of This Utility. This Port Consists **Of DIFF, DIF3, CMP & SDIF.** A copy of **ELVIS [SJS 46]** is also required.

SJS 104:- **UNGIF Release 091** This Is The Latest Version Of This Gif Viewer / Manipulation Program. This Version Requires The Pointer Environment To Work. It Is Compatible With Most GIF89 File Formats.

SJS 105:- **QL BBS MESSAGE READER.** With This Program You Can Capture A Group Of Messages From A BBS, Log Off ,Then Read The Messages With This Utility.

SJS 106:- **R.C.S. Revision Control System.** This Utility Manages Multiple Revisions Of Files. It Is Very Useful For Text Files That Are Revised Frequently.

SJS 107:- **6 DISKS OF CLIP-ART.** This Series Of Clip-Art Covers Religious, Seasonal & Holidays. It Is The Work Of DILWYN JONES.

SJS 108. **SHOW TIA.** This Is A Small Utility To Show Clip-Art Files Ported From The TI-99/4A Computer. It Contains A Collection Of Clip-Art Already Ported Over.

SJS 109. **ROCK 'N ROLL.** This Is A Selection Of Music Based Clip-Art Files For Use With LineDesign. The Images Are Of Use To Set Designers Working In The Music Industry.

SJS 110. **SCHEME INTERPRETER.** This Is A 2 Disk Package That Contains The Full Scheme Interpreter For The QL. The Package Includes The Exe Program, Source And Manuals.

SJS 111. **APL INTERPRETER.** APL Is A New Interpreter To The QL. The Full Package Including Source, Manuals And Executable Interpreter Comes On 1 Disk.

SJS 112. **INFOCOM.** With This Utility You Can Play PC Infocom ™ Text Adventure Games On The QL. A Selection Of Infocom Compatible Data Files Are Included In The Package.

SJS 113. **TEXT 87 PLUS 4 DEMO.** This Disk Contains A Full Working Demo Version Of This Popular Word Processor By Software 87.

SJS 114. **EXPANDING EARTH VERSION 1.08.** This Program Is A Graphical Representation Showing The Movements Of The Earth's Continents (Continental Drifts) From 200 Million Years Ago To Present Day.

PUBLIC DOMAIN

SJS 115. **COOPER CLIP ART COLLECTION.** A Three Disk Package Of Cooper Clip Art In Compressed Format. Each Disk Contains A Screen Viewer And A Conversion Program To QL Screens.

SJS 116. **QPOINT VERSION 2.01.** Erik Slagter's BBS Utility. This Program Makes Life As A Point Easier.
Requires Pointer Environment.

The classic books section has also been updated to include the following titles:-

SJPD CB68:- **Buddha,**
His Life & Teachings

SJPD CB69:- **Confucius.**
The Great Learning's.

SJPD CB70:- **The Holy Koran. 3 Disks**

SJPD CB71:- **Edgar Rice Burroughs.**
Tarzan And The Jewels Of Opar.

SJPD CB72:- **The Deuterocannonical Books Of The Bible. Part 1.**

SJPD CB73:- **The Deuterocannonical Books Of The Bible. Part 2.**

SJPD CB 74:- **Susan Rowson.**
Charlotte Temple.

SJPD CB 75:- **Oscar Wilde**
The Picture Of Dorian Gray..

SJPD CB76:-. **Gene Stratton Porter.**
A Girl Of The Limberlost. Part 1

SJPD CB77:- **Gene Stratton Porter.**
A Girl Of The Limberlost Part 2.

SJPD CB78:- **Guston Leroux.**
The Phantom Of The Opera

SJPD CB79:- **Thomas Hardy.**
Tess Of D'urbervillees. Part 1

SJPD CB80:- **Thomas Hardy.**
Tess Of D'urbervillees. Part 2.

SJPD CB81:- **James M. Barrrie.**
Peter Pan.

SJPD CB82:- **Star Trek Generations.** This Is The Full Script To The NEW Film In The Star Trek Series.

SJPD CB83:-
Aladdin And His Magic Lamp
A Christmas Carol.
A Child's Garden Of Verse.

SJPD CB84:- The Book Of Mormon.
A 3 Disks Package.

SJPD CB85: Robinson Crusoe.

SJPD CB86: Jane Eyre. Part 1.

SJPD CB87: Jane Eyre. Part 2.

SJPD CB88: Ch'an Tao Chia Essays & Lectures.
Tao Te Chang.
Meditation.

SJPD CB89: Belisarious & Neurophyre's
Hack Facts For 1994.
The Official Phreakers Manual.
THIS DISK IS FOR INFORMATION ONLY.

There has also been a few updates to existing disk.

C68 has now been updated to version 4.14c.

Q.T.P.I. is now at version 1.45a

Lib Curses for C68 is updated to version 1.23.

As you can see I have been rather busy in the past few months. To give a full description of the new disks would fill **QREVIEW** and MAY upset Bruce! A full description can be found on my catalogue disk which is *FREE* if you send a formatted QL disk and return postage.

It very nice to see *SOLVE-IT PLUS* has now been published. This disk originally submitted to **S.J.P.D. SOFTWare** as a public domain program, I was so impressed with the very high quality of the program. I sent Dilwyn Jones a copy to see if he would be interested in marketing such a program. He would, He did & and now the author is now earning money. It can pay to submit programs to **S.J.P.D. SOFTWare!**

Keep sending the programs for evaluation you never know, you could be next on some pay roll.

GOOD LUCK
Steve Johnson.
MARCH 1995.

Quo Vadis DESIGN

Computer Consultancy/Services
Proprietor: Bruce Nicholls
57 Shaftesbury Road, Romford,
Essex, RM1 2QJ, UK.
Tel/Fax: (01708) 755759

QL SOFTWARE

GRAPHICAL

Scanned Clipart 1&2	£10
Vision Mixer 1	£10
Image Processor	£15
Convert-PCX	£10
The Clipart	£12
The Painter	£25
QRactal	£20
ST-O-QL	£12.50
3D Terrain	£12.50
Quick Mandelbrot 3	£15.00
Image-D	£10
Sidewinder Plus QL	£24.95
Open World	£18

TEXT

Banter	£15
Sidewriter	£15
DeskJet A5	£12
Quick Posters	£10
Text87 Plus4	£79
2488 printer drivers	£19
Typeset94 Deskjet driver for Plus4	£29
Fountext94 + Founted89 for Plus4	£39
Publisher Pack driver for Plus4	£30

(Drivers for use with Linedesign 2)

UTILITIES

Screen Dazzler	£10
Dev Manager	£15
Super Disk Labeller	£10
The Cat	£5
QL-PC Fileserver II	£35
Files 2	£12
TaskMaster	£25
TaskMaster Hard disk	£35
Spellbound	£30
Spellbound S/E	£50
Cueshell	£40
QPAC1	£19.95
QPAC2	£39.95
QTYP2	£29.95
4Matter	£14.95
Locksmithe	£14.95
Toolkit 2 (disk)	£19.95
Floppy Disk Utilities	£18

GAMES/SIMULATION

5 Game Pack	£12.50
Grey Wolf	£12.50
Open Golf	£12.50
Return to Eden	£17.50
Quiz Master 2	£10
The Fugitive	£9.95
Fleet Tactical Command II	£39.95
FTCII Data Print Utility	£9.95
Flightdeck	£15
Spectrum Emulator ZM/128	£28
Spectrum Emulator ZM/HT	£40

PROGRAMMING

Basic Reporter	£10
QLiberator V3.36	£50
QLiberator budget version	£25
QLOAD & QREF	£15
Resident Program Manager	£10
EASYPTR 3 Part 1	£40
EASYPTR 3 Part 2	£20
EASYPTR 3 Part 3	£20
DISA 2	£40
DEA disassembler	£26
MasterBasic	£22
Q-Library Manager	£18

MISCELLANEOUS

Amd-Airplan	£10
Flashback	£25
Flashback S/E	£40
Music Manager	£12

TERMS/CONDITIONS

All prices shown are in UK pounds Sterling. Software is sent post free in the UK, overseas add £1.00 per order. Please make payments payable to 'Quo Vadis Design'. Payment in UK pounds Sterling currency only, Cheques (drawn on a UK branch of a bank or building society), Postal Orders, International Postal Orders and Eurocheques are all accepted. Goods remain the property of Quo Vadis Design until full payment has been received. Call or write for a more comprehensive catalogue.